NCT-MINICUBE 78K0

快速入门手册

VERSION V1.20

NEC Electronics (China) Co.,Ltd
NEC Electronics Hong Kong Limited

Date Published　Feb，2006

本商品、技术或软件的出口必须依据出口国的出口管理规章来进行。不能违反当地法律。

前言

目的

本手册的目的是向用户讲解 NCT-MINICUBE78K0 开发工具的基本概念和操作步骤。通过本手册的指导，用户可以初步掌握 NEC 的 Applilet，PM+和 ID78K0-QB for MINICUBE 等软件工具的使用。结合 NCT MINICUBE 以及自带的仿真板等硬件，用户可以很方便的实现目标系统，迅速掌握 NEC 78K0 系列芯片的开发流程。

预备知识

本手册面向的是准备使用 NEC 78K0 系列芯片进行开发的初学者。读者需要具备一定的 C 语言、汇编语言和微控制器（MCU）的基础知识。

章节

本手册分为以下几个章节：

- 工具安装
- 应用范例
- 工具简介
- 代码生成
- 编译修改
- 硬件连接
- 仿真调试

术语

78K0：NEC 的 78K 系列 8 位 MCU 产品分为 78K0 和 78K0S 子系列。这两个子系列 MCU 分别有各自对应的开发工具。

参考文档

本手册参考的文档都是NEC正式发布的文档，可以在以下网址索引到：
http://www.necel.com/en/search/index.html#document

参考文档

名称	编号
CC78K0 Ver3.70 C Complier Operation User's Manual	U17201E
RA78K0 Ver3.80 Assembler Package Operation User's Manual	U17199E
ID78K0-QB Ver2.90 Integrated Debugger Operation User's Manual	U17437E
PM plus Ver5.20 User's Manual	U16934E
NCT-MINICUBE78K0 On-Chip Debug Emulator User's Manual	U17029E
78K0,78K0S Series OCD Checker V1.02 User's Manual	ZUD-CD-05

目 录

插图目录

表格目录

第 1 章　开发工具的安装

本章将介绍如何安装 NEC 工具包 Applilet、RA78Kx/PM+、CC78Kx 和 ID78K0-QB for MINICUBE。

1.1　驱动代码生成工具 Applilet 的安装

1.1.1　Applilet 简介

Applilet for 78K0KX2 是帮助初次接触 NEC78K0 系列 MCU 的初学者快速入门生成驱动代码工具。用户可以绕过冗长的手册阅读，而直接快速生成自己的底层驱动和代码框架。在生成的代码基础上稍作修改就可以得到高质量的程序代码。

Applilet for 78K0KX2 生成的 C 代码符合 ANSI 标准 C，还可以选择生成代码为汇编格式。Applilet 生成的程序文件和头文件按照模块独立存放，结构清晰。

Applilet for 78K0KX2 包括图形化的参数配置界面，可以方便的查看各个引脚的占用情况，如果管脚复用冲突，会自动提示。通过 Applilet，可以速查各个寄存器的内存地址、预览程序和方便的观察。更多详细的介绍在本手册第 4 章。

目前正式发布的适用于 78K0 系列的 Applilet 版本为：V1.51。

1.1.2　安装步骤

运行 Applilet for 78K0KX2 安装包，开始安装。

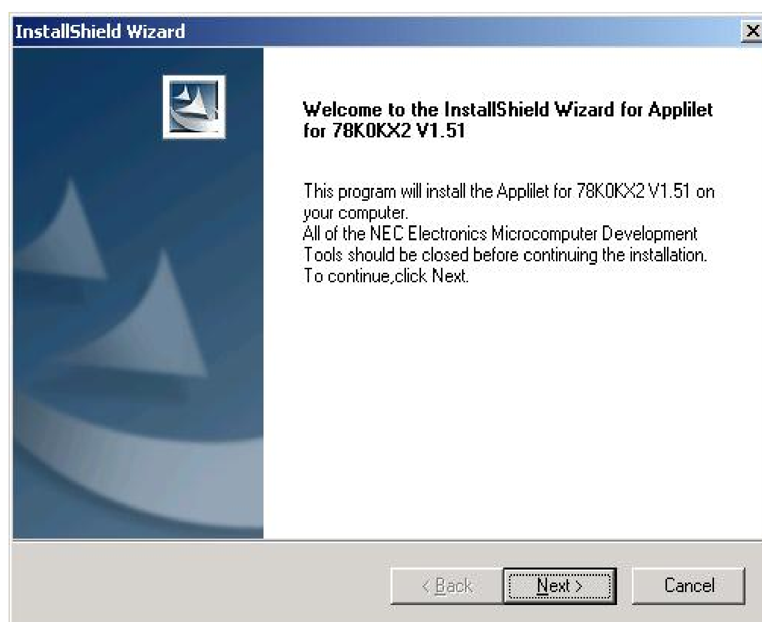


图1-1　Welcome 对话框

点击“Next 按钮”，出现 License Agreement 对话框。

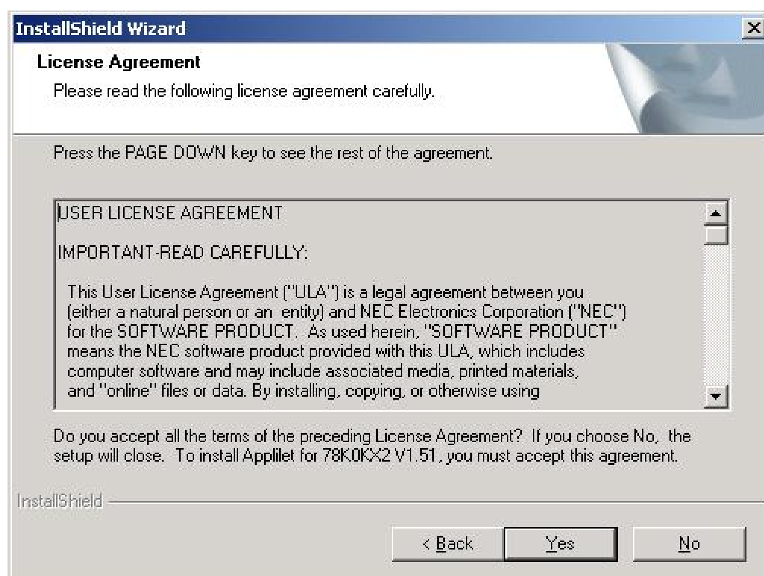


图 1-2　License 对话框

接下来选择安装目录和程序目录，安装结束提示如下信息：

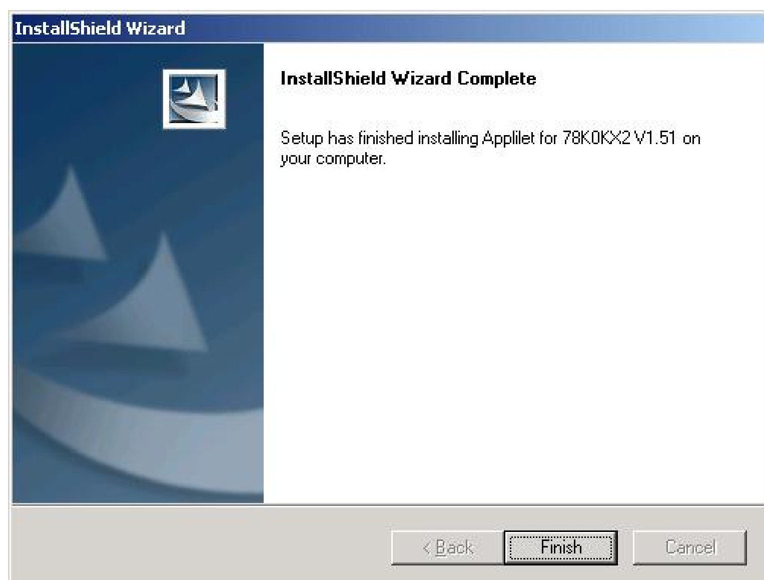


图 1-3　Applilet 安装完成

到此，Applilet 的安装就完成。启动 Applilet 请点击“开始→程序→NEC Electronics Tools→Applilet for 78K0KX2”。

1.2　汇编器 RA78Kx 以及集成开发环境 PM+的安装

1.2.1　安装包内容

RA78Kx 安装包中包括 RA78Kx 和 PM+ 78Kx 两个软件包：

RA78Kx：

RA78Kx 是用于 NEC 78Kx 系列微处理器的汇编编译工具包。它将汇编源程序编译为可以被 78Kx 芯片识别的目标代码。

PM+：

PM+是一个集成开发环境平台，用来有效地开发 NEC 的 8/16 位微控制器 78Kx 系列的用户程序。PM+包括一个项目管理器和一个编辑器，提供了一系列的操作功能，如编辑器功能、编译器功能、开发向导功能等。

本软件同样适用于 NEC32 位的 V850 系列 MCU 的开发。

1.2.2　安装步骤

运行 RA78Kx 安装包中的“setup.exe”文件，开始安装。

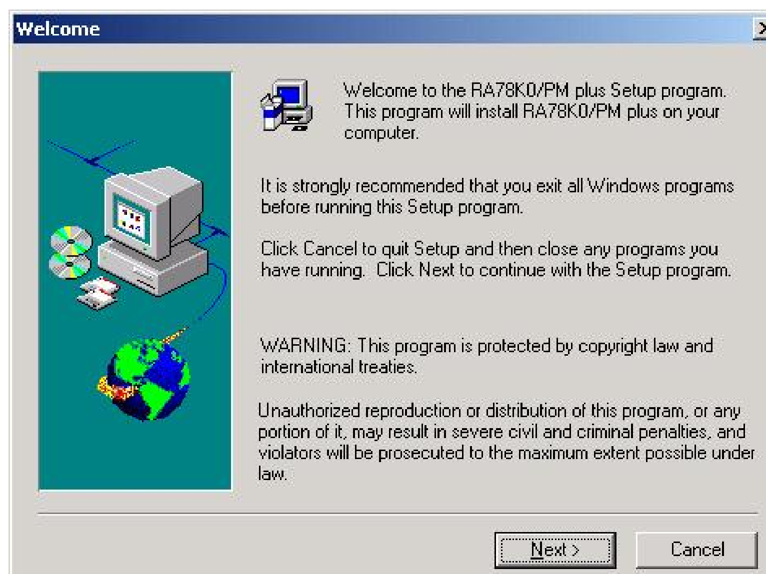


图 1-4　Welcome 对话框

点击“Next”按钮，出现“Software License Agreement”对话框。

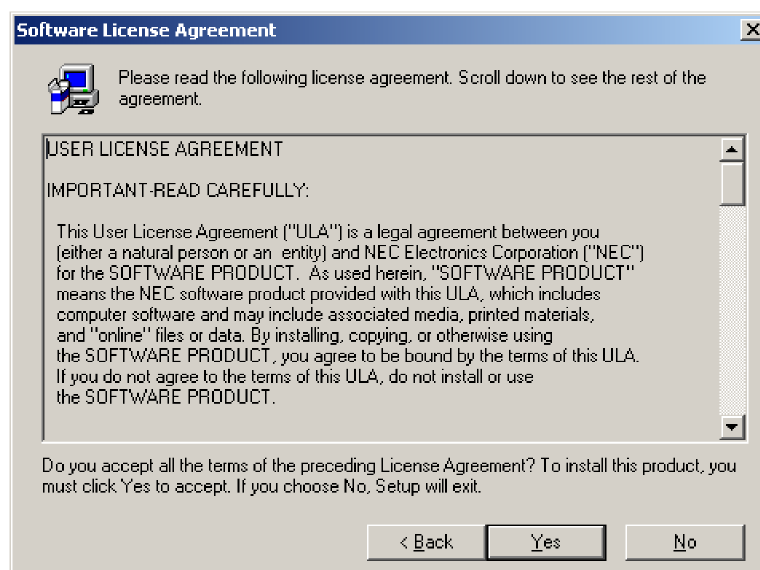


图 1-5　Software License Agreement 对话框

点击“Yes”按钮，出现“Product ID”对话框。



图 1-6　Product ID 对话框

输入安装光盘上的 9 位产品序列号，点击“Next”按钮，出现“Select Components”对话框。选择要安装的组件，默认选择 RA78Kx Assembler Package 和 PM+以及它们的相关文档。

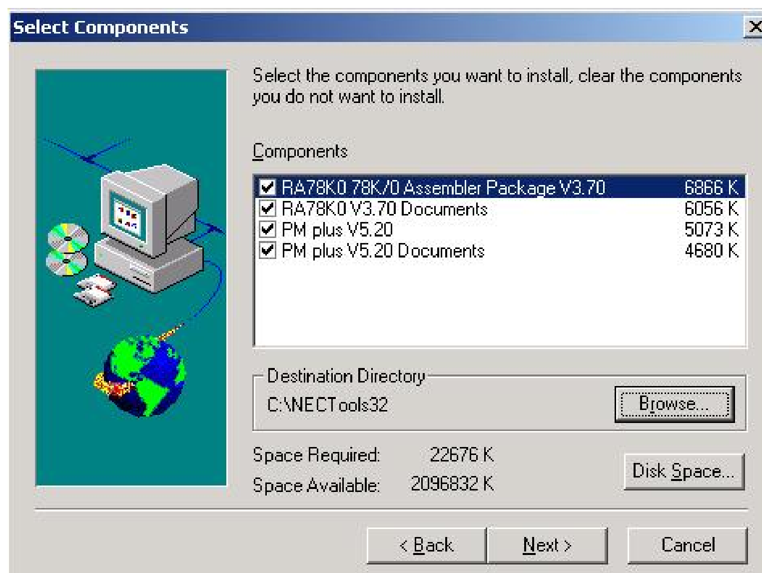


图 1-7　Select Components 对话框

点击“Browse”按钮，指定要安装的路径，默认路径是“C:\NECTools32”。点击 Next，出现“Select Program Folder”对话框，默认设置是“NEC Tools32”，用户可以自行修改。

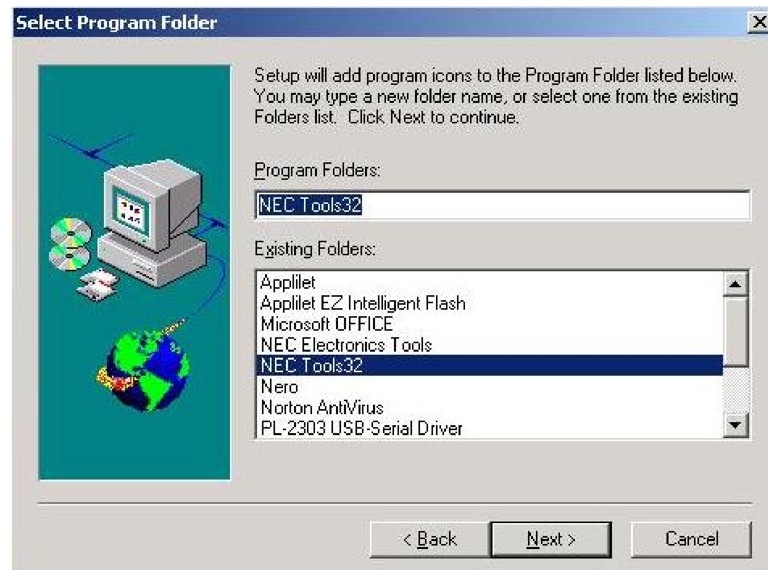


图 1-8　Select Program Folder 对话框

点击“Next”，出现“Start Copying Files”对话框。检查当前的设置。

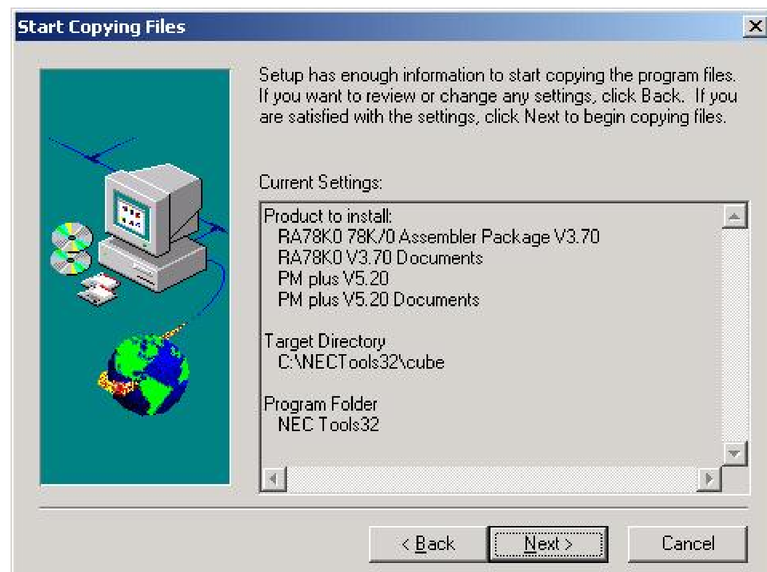


图 1-9　Start Copying Files 对话框

点击“Next”开始拷贝文件，等待拷贝完成。

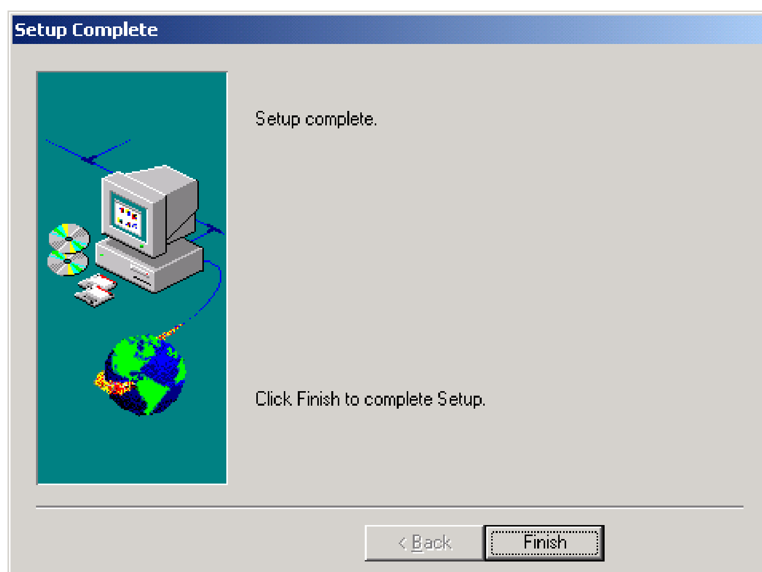


图 1-10　Setup Complete 对话框

出现 Setup Complete 对话框，点击“Finish”按钮，安装就完成了。

1.3　C 编译器 CC78Kx 的安装

1.3.1　安装包内容

CC78Kx 是用于 NEC 78Kx 系列微处理器的 C 语言编译工具包。它具有以下特征：

- 符合 ANSI 标准的 C 语言库
- 支持程序 ROM 存储方式
- 集成多种编译功能，灵活易用

1.3.2　安装步骤

运行 CC78Kx 安装包中的“setup.exe”文件，开始安装。

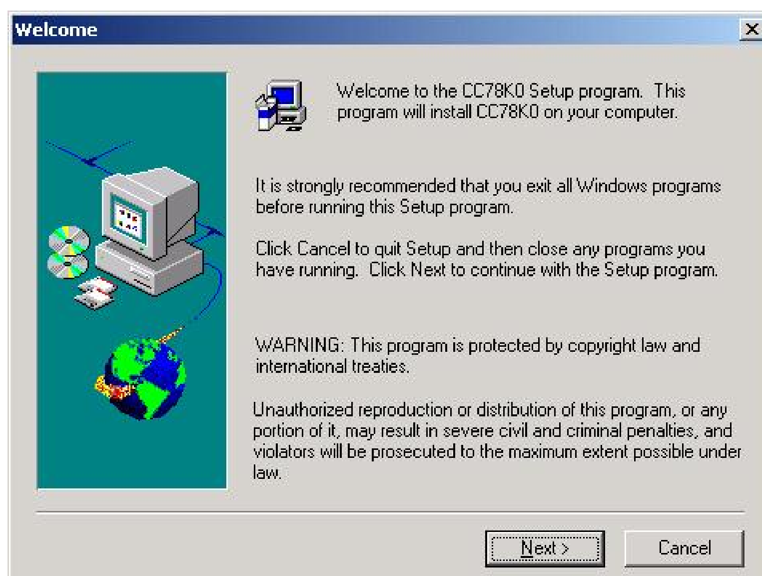


图 1-11　Welcome 对话框

点击“Next”按钮，出现“Software License Agreement”对话框。

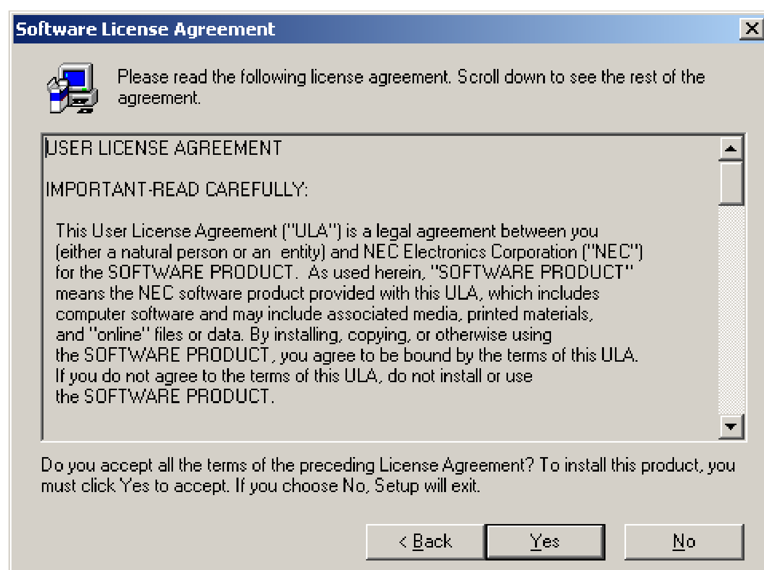


图 1-12　Software License Agreement 对话框

点击“Yes”按钮，出现“Product ID”对话框。



图 1-13　Product ID 对话框

输入产品序列号，点击“Next”按钮，出现“Select Components”对话框。选择要安装的组件，默认选择 CC78Kx C Compiler 和它的相关文档。

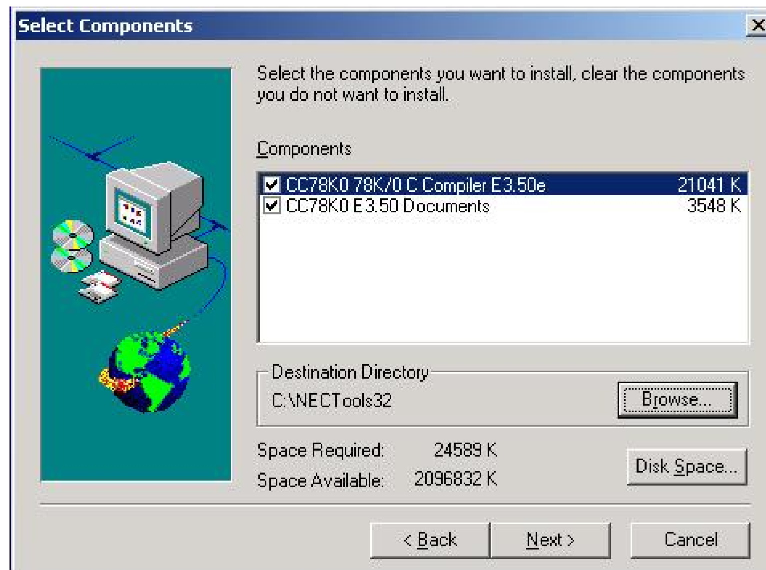


图 1-14　Select Components 对话框

点击“Browse”按钮，指定要安装的路径，默认路径是“C:\NECTools32”。点击 Next，出现“Select Program Folder”对话框。默认设置是“NEC Tools32”，用户可以自行修改。

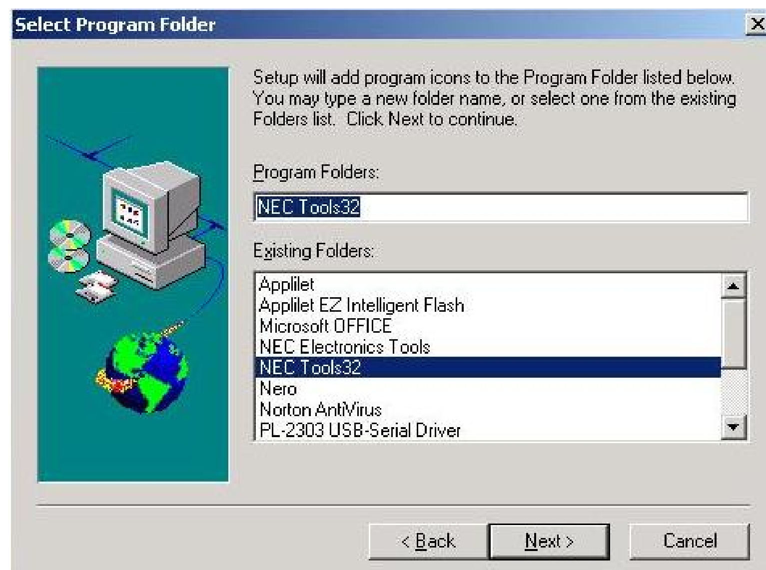


图 1-15　Select Program Folder 对话框

点击“Next”，出现“Start Copying Files”对话框，检查当前的设置。

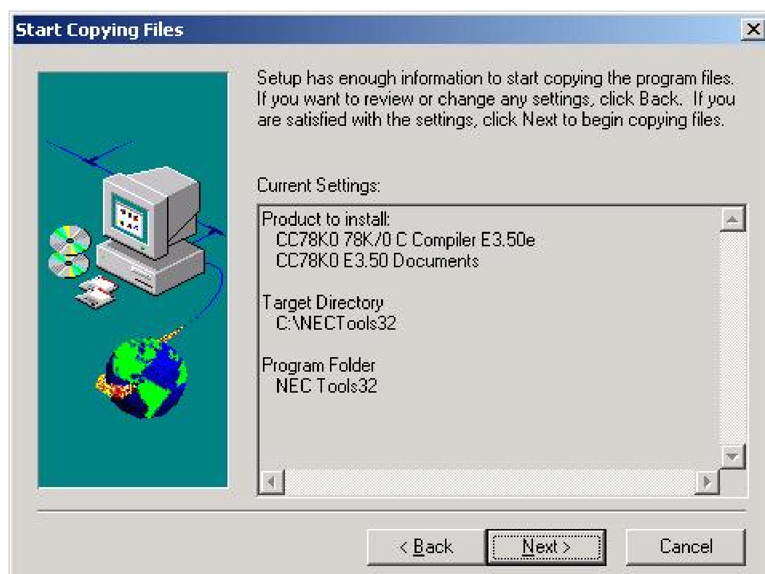


图 1-16　Start Copying Files 对话框

点击“Next”开始拷贝文件，等待拷贝完成。

1.4　硬件调试工具 ID78K0-QB for MINICUBE 的安装

1.4.1　安装包内容

ID78K0-QB for MINICUBE 是 NEC 提供的调试软件，来帮助用户调试程序，观察程序执行结果。它可以对 NEC 的多种 CPU 和芯片进行调试，给用户提供了一个集仿真、调试、性能分析于一体的功能强大的软件平台。

1.4.2　安装步骤

执行 ID78K0-QB for MINICUBE 的“Setup.exe”程序，出现“Welcome”对话框。



图 1-17　Welcome 对话框

点击“Next”按钮，出现软件许可协议对话框。

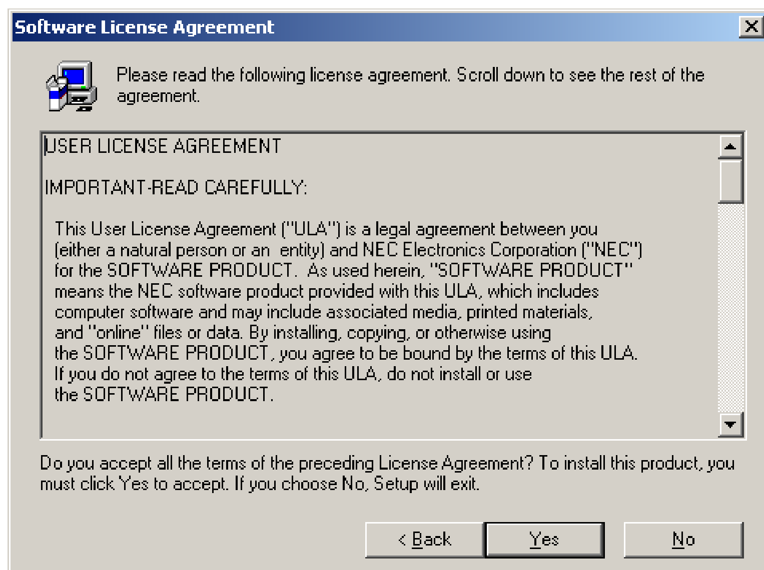


图 1-18　Software License Agreemen 对话框

点击“Yes”按钮，出现“Product ID”对话框。



图 1-19　Product ID 对话框

输入 MINICUBE 软件安装光盘上的序列号，点击“Next”按钮，出现选择组件对话框。



图 1-20　Select Components 对话框

选择要安装的组件。

点击“Browse”按钮，指定要安装的路径，默认路径是“C:\NECTools32”。点击“Next”，出现 Select Program Folder 对话框。默认设置是“NEC Tools32”，用户可以自行修改。

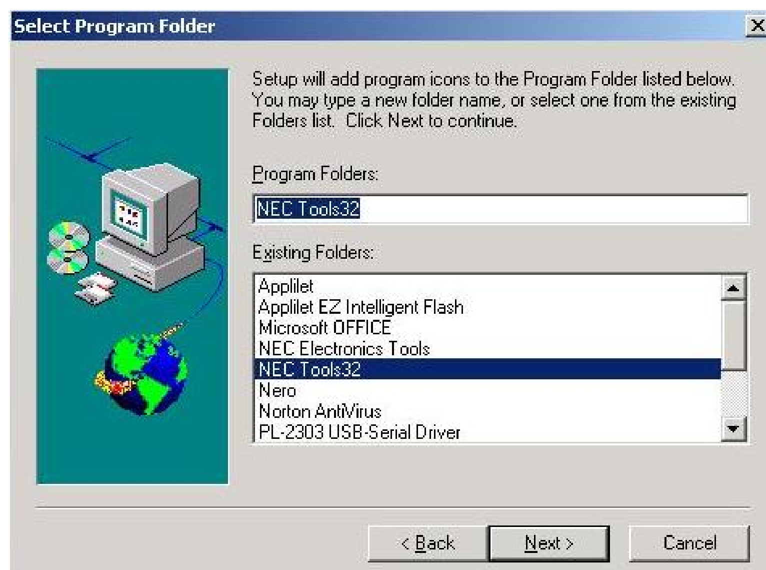


图 1-21　Select Program Folder 对话框

点击“Next”，出现“Start Copying Files”对话框，检查当前的设置。

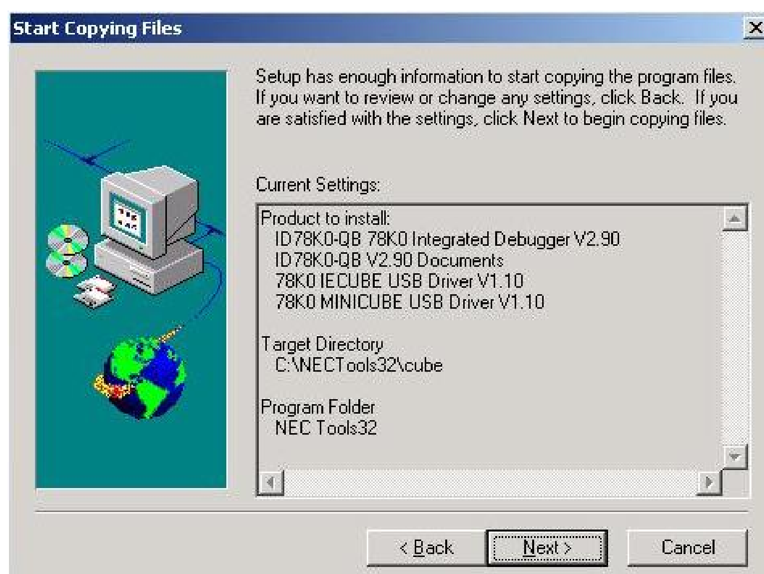


图 1-22　Start Copying Files 对话框

点击“Next”开始拷贝文件，等待拷贝完成。

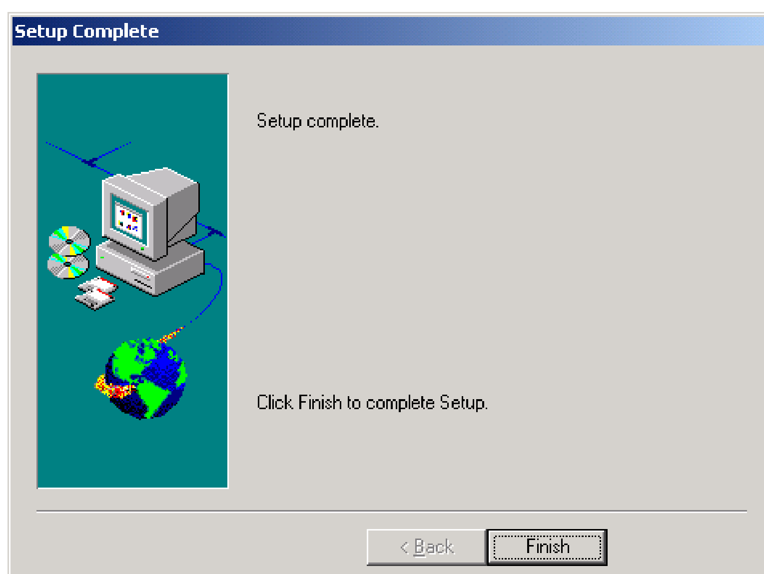


图 1-23　Setup Complete 对话框

出现“Setup Complete”对话框，点击“Finish”按钮，安装就完成了。

1.5　设备文件的安装

1.5.1　设备文件简介

设备文件（Device file）是描述芯片内部信息的二进制文件。每一个设备文件描述了一个或一组芯片的配置信息。

在编译程序的过程中，编译工具需要访问设备文件来获得相应芯片的内存和寄存器信息。

在用 PM+编译程序的过程中，PM+也需要访问设备文件来获取芯片的配置信息。

用户可以到以下网址下载 78K0 系列各个芯片的设备文件：

http://www.necel.com/micro/ods/eng/tool/DeviceFile/78K0.html

1.5.2 安装步骤

点击 “开始→程序→NEC Tools32→Device File Installer”图标，打开“Device File Installer”工具。

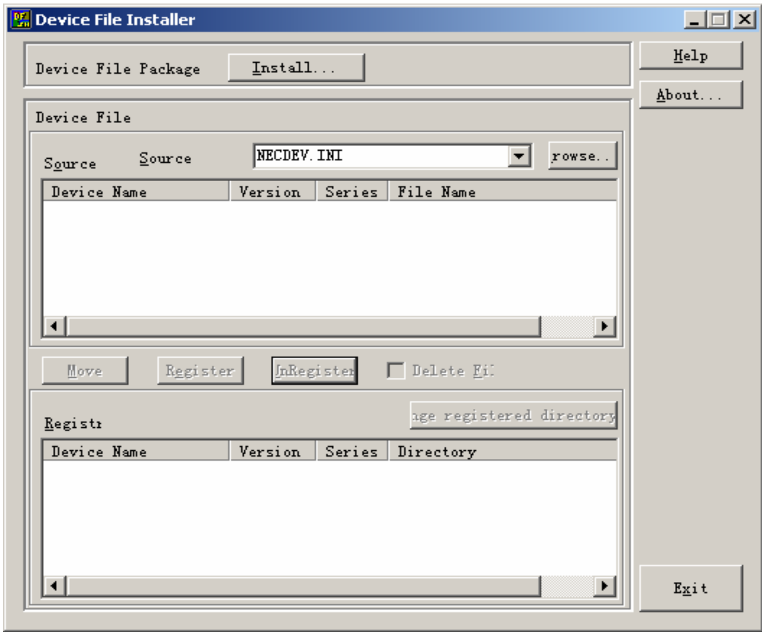


图 1-24 Device File Installer 界面

点击“Browse”按钮，选择设备文件（.78k）所在的路径。默认目录为 NECTools32 目录下的 dev 目录，也可以是已经下载的设备文件所在的存放目录。

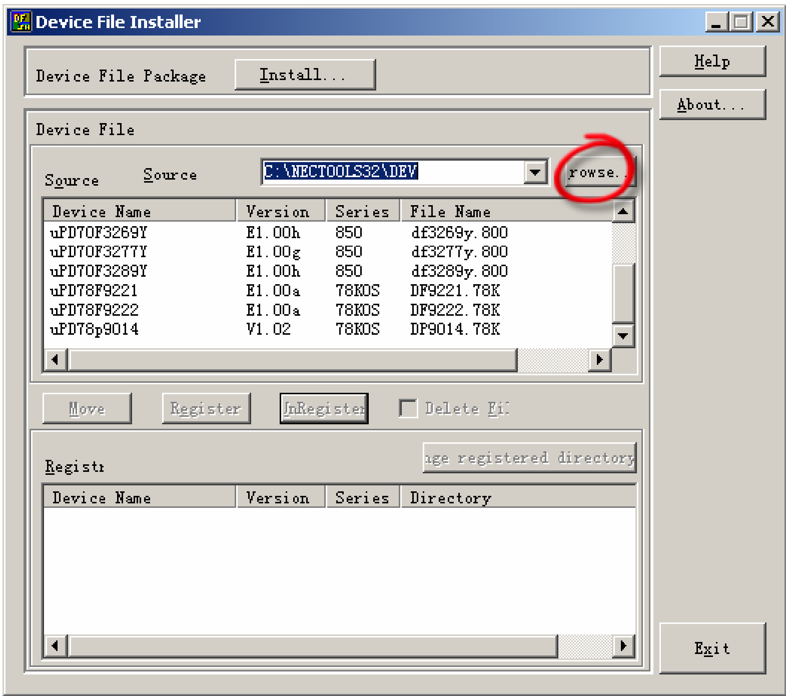


图 1-25 选择设备文件路径

在列表中选中一个或多个设备文件。

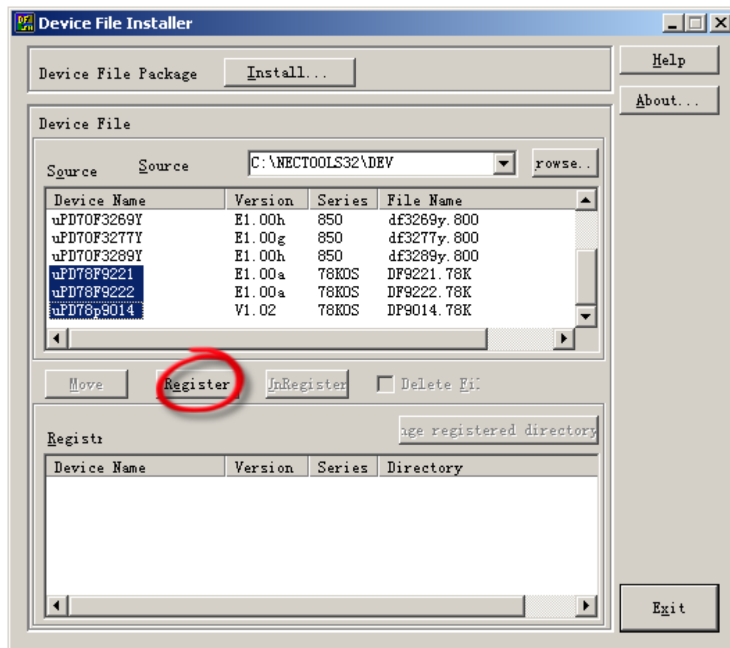


图 1-26　选择设备文件

点击“Register”按钮，执行安装。

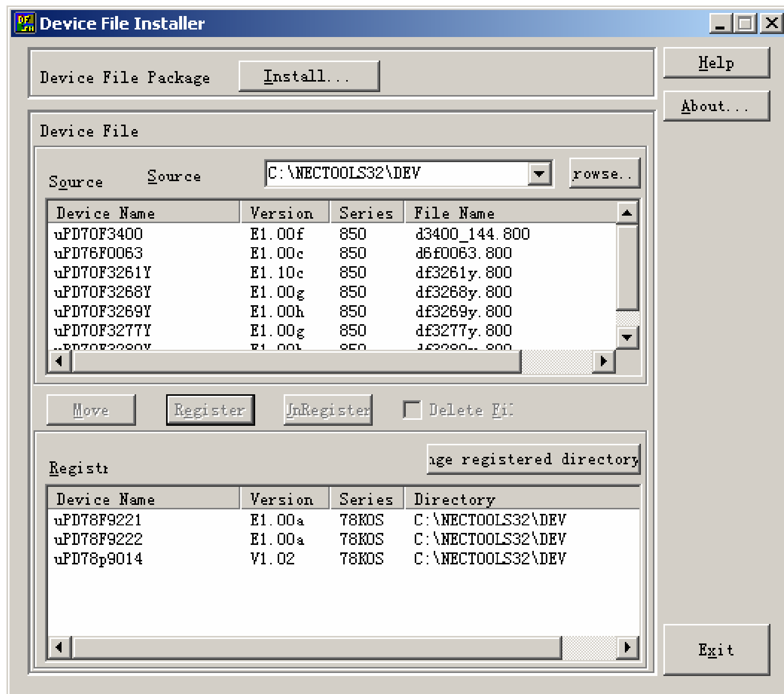


图 1-27　设备文件安装完成

至此，所有需要的软件和相关支持文件都已经安装完毕，下一章我们介绍一个 NEC 78K0/KF2 的应用例程。

第 2 章　应用范例

2.1　软硬件环境

本 Demo 程序使用的芯片是 78K0/KX2 系列的μPD78F0547D。请参考手册《78K0/KF2 8-Bit Single-Chip Microcontrollers Preliminary User’s Manual》（U17397E），以获得该芯片的更多信息。

范例程序专为演示 Applilet，PM+和 ID78K0-QB for MINICUBE 软件工具的操作而创建。程序驱动代码的生成需要 Applilet for 78K0/KF2 1.50 以上版本，编译 Demo 需要 PM+ 5.10 以上版本，调试 Demo 需要 ID78K0-QB for MINICUBE 2.81 以上版本。

范例程序使用的硬件工具包括 NCT-MINICUBE78K0 以及产品包装中自带的 USB 连接线、10 针排线、仿真板（芯片型号为μPD78F0547DGC）和诊断板。

2.2　驱动代码的生成

2.2.1　Applilet 的启动

点击Applilet图标，就会进入 Applilet 的应用程序界面，点击“File→New”，出现如下的对话框，此处我们选择的芯片为 78K0/KF2 系列的μPD78F0547_80，也就是 MINICUBE 自带的仿真板上的 MCU 型号。

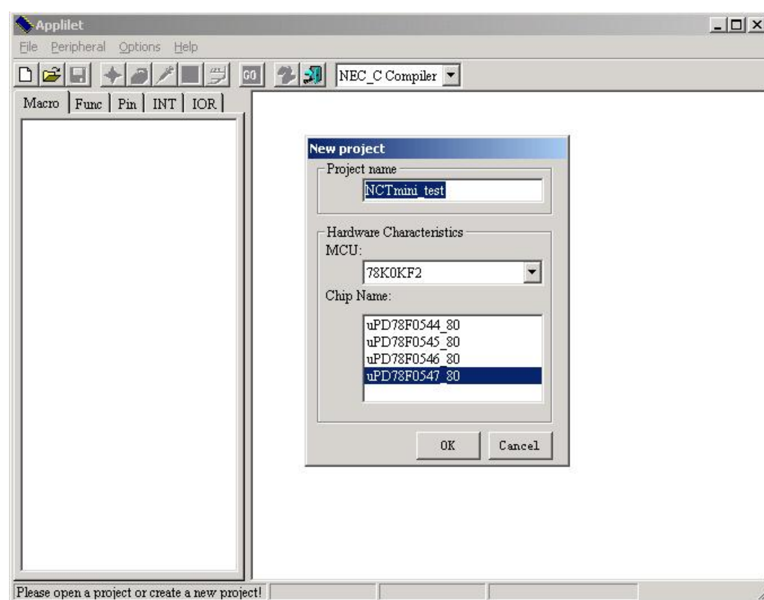


图 2-1　在 Applilet 中选择目标芯片

图 2-1 中的“Project name”处可以自由命名，如果不做修改默认为 mdt（multi driver tools），生成的 HEX 文件也将默认使用这个名称。

芯片选择结束后，点击“OK”按钮就进入了如下的 Macro 界面：

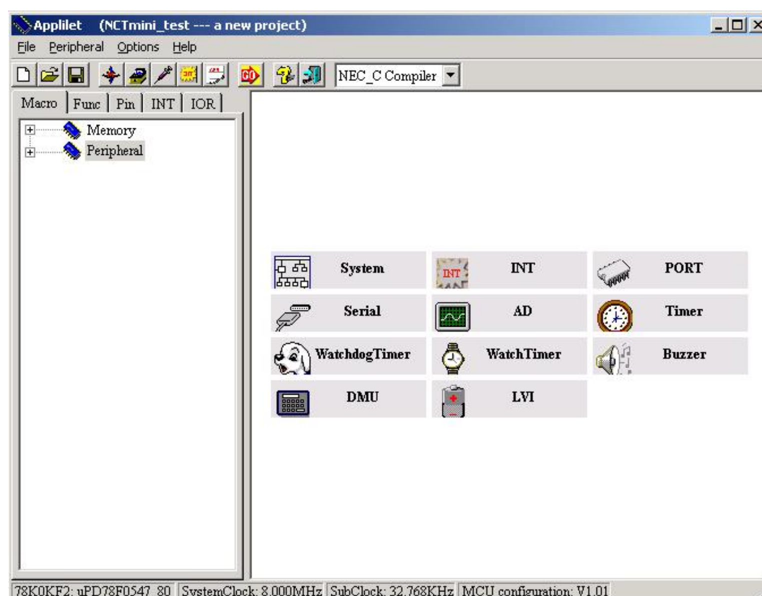


图 2-2　Applilet 的 Macro 界面

例程的功能：是控制诊断板上自带的两个 LED 灯的亮灭闪烁。控制 LED 的亮灭，需要从端口引脚提供高低电平；控制闪烁的频率，就要设置定时器中断的间隔；从 PC 机上调试程序，需要使用 OCD 功能。所以接下来对这三项功能在 Applilet 中进行配置。

注：OCD 功能要求目标板上的 MCU 必须是支持调试功能的 D 版本。比如调试的时候使用的是 μPD78F0547D 版本的 MCU，那么将来目标板使用 μPD78F0544，μPD78F0545，μPD78F0546，μPD78F0547 四款 MCU 都是可以支持的。

2.2.2　Port 的配置

先看端口的设置，点击图 2-2 中的“PORT”图标，就看到下图的设置界面。诊断板上 LED 连接的是芯片 μPD78F0547D 的 P32，P33 端口。

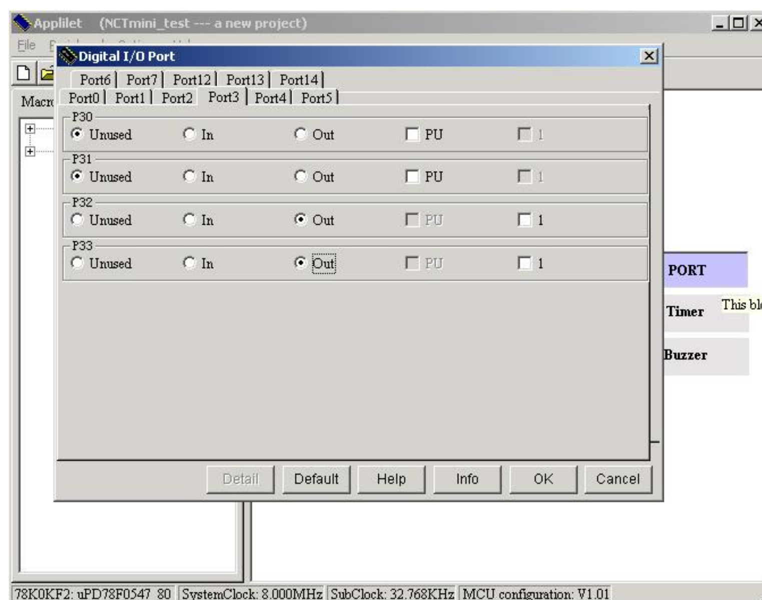


图 2-3　Applilet 的 Port 的设置界面

设置 Port 的时候，如果各个功能占用的相关资源发生冲突，Applilet 会自动给出提示。

2.2.3　Timer 的配置

Applilet 软件中“Timer”模块的设置如下图所示，为简便起见，使用的是定时器 TM00 最简单的 Interval 功能。

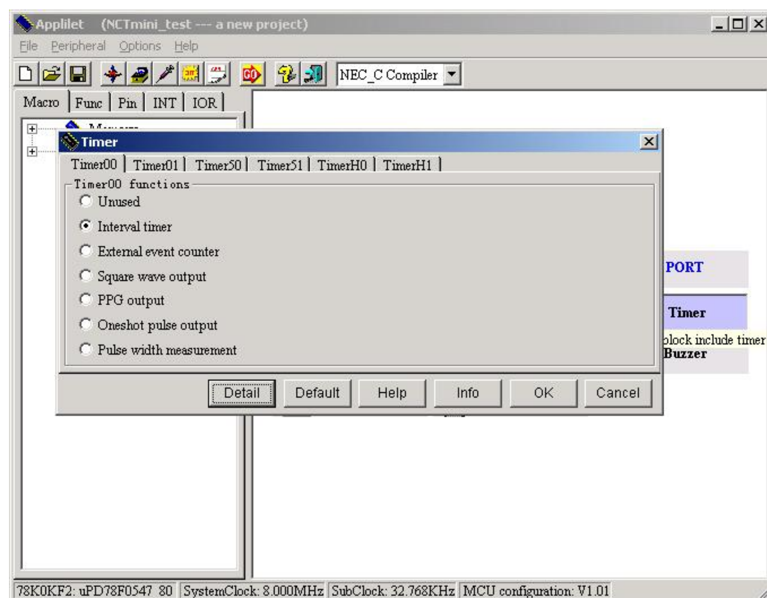


图 2-4　Applilet 的 Timer 的模式设置界面

在上图中点击“Detail”按钮，就会看到定时器的详细设置。

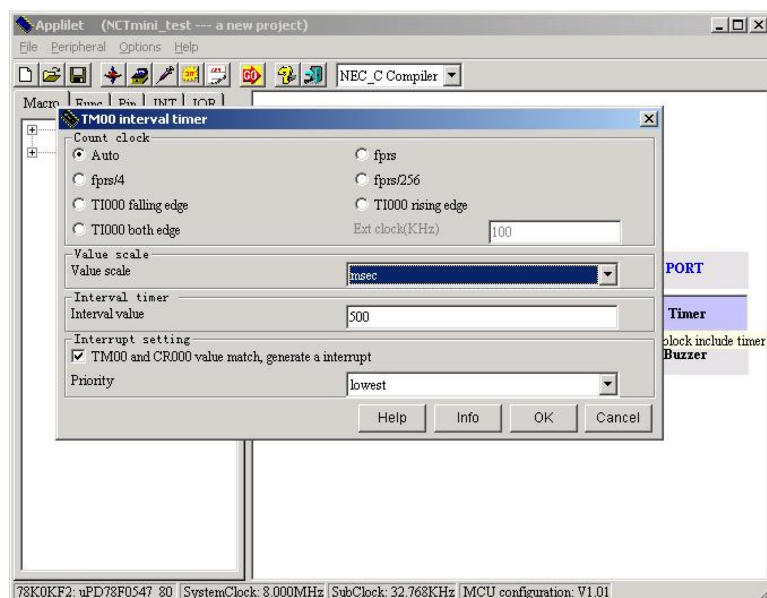


图 2-5　Applilet 的 Timer 的详细参数设置界面

在上图中我们选择的闪烁频率是 2Hz，也就是 0.5 秒动作一次，所以在 Value scale 中选择“msec”，在 interval value 中输入“500”。Applilet 软件会按照当前时钟和你要求的时间算出应该取的时钟计数值放入比较器，激活 Interrupt setting 中的“TM00 and CR000 value match，generate a interrupt”的复选框。然后点击“OK”，Timer 设定完毕，回到图 2-2 的 Macro 界面。

2.2.4 System 的配置

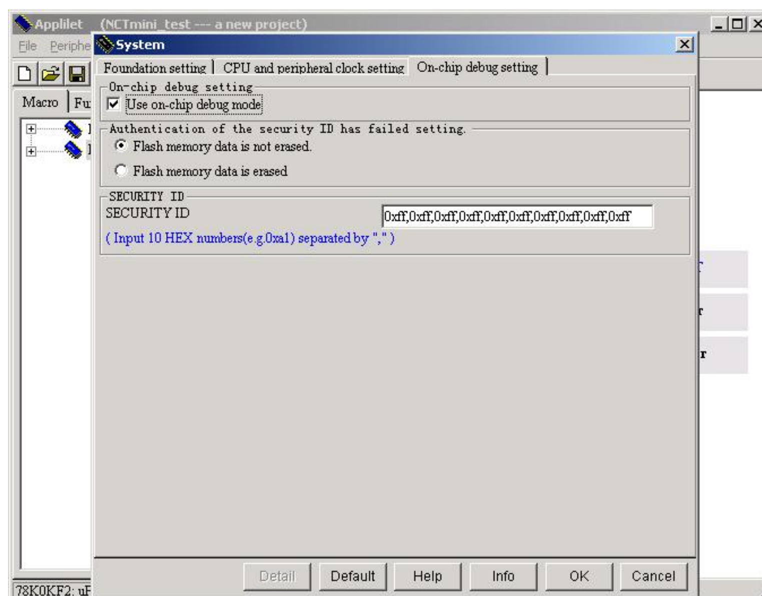


图 2-6 在 Applilet 中开启 OCD 功能

在图 2-2 中点击“System”图例，就会进入图 2-6 的界面，在“On-chip debug setting”标签下选中“Use On-chip debug mode”，就开启了片上调试功能，可以让我们方便的通过 PC 来观察或修改目标板上的用户程序运行的详细内容。

注意：

1. 如果在 Applilet 中没有打开 OCD 功能，那么在 PM+中程序编译链接时不会有问题，但是在 ID78K0-QB for MINICUBE 中下载时就会出现下面的错误。

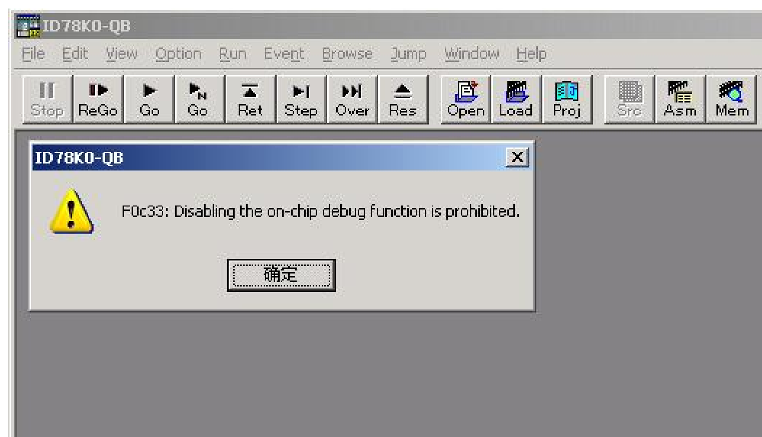


图 2-7 未设置 OCD 功能导致无法进入调试器

2. 在图 2-6 中的 SECURITY ID 部分，默认为原来的 10 个 0xFF，如果想对自己的程序保密，那么请输入自己的密码，此处请牢记密码，否则程序通过 MINICUBE 下载后，下次将无法进入调试界面，出现三次密码输入错误，则 ID78K0QB for MINICUBE 程序自动关闭。
3. 如果没有打开 OCD 功能，那么用户程序也可以使用编程器直接下载到目标 MCU 运行，但是无法调试，也就是不能使用 ID78K0QB for MINICUBE 或者 IECUBE 软件从 PC 上来观察程序的运行，即不能对程序进行调试和修改。

2.2.5 生成代码

其它的选项一般不需要更改，以上三个模块设置完成后，程序的框架就大概设置好了，如果需要添加更复杂的功能，可以参阅相关产品手册。

参数设置完成之后，点击 Applilet 主界面菜单栏上的“GO”按钮，就进入了如下的程序生成对话界面：

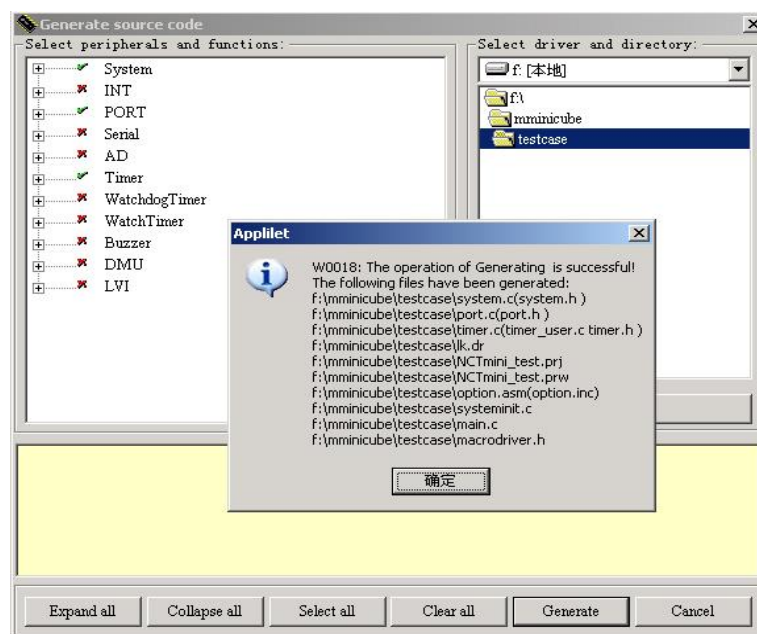


图 2-8 Applilet 程序文件的生成

在右边的“Select driver and directory”中选择自己希望存放 Applilet 生成的程序文件的文件夹，点击右下方的“Generate”就开始生成程序并存储。完成后会弹出一个报告框，列出所有已经生成的文件以及存放的位置。

2.3 编辑、编译、链接

使用了 Applilet 工具软件生成程序文件之后，用户可以使用 PM+直接载入生成的程序文件，操作如下：

用 PM+ 打开工程文件 NCTmini_test.prw。如果不是第一次打开 PM+时，那么总是会自动打开上一次关闭时正处于工作中的工程文件（也可以在 PM+中选择下次启动时不再自行载入）。

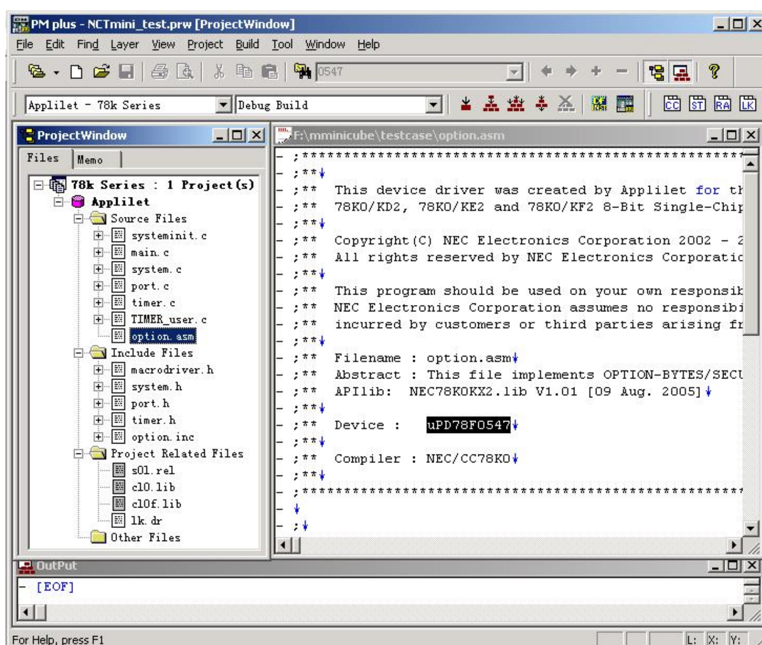


图 2-9　用 PM+打开工程

此工程中，因为 Applilet 生成的只是程序的驱动框架，还需要添加一点自己的代码。首先要在主程序中启动定时器 TMM，添加代码如下：

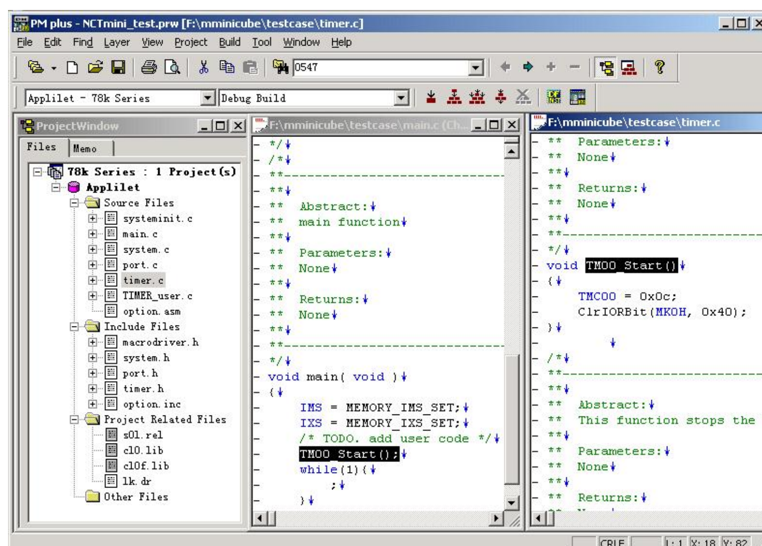


图 2-10 定时器 TM00 的使能

1. 在主程序文件 main.c 的主函数中添加“TM00_Start（）;”启动定时器。位置在主程序初始化完成之后。

   备注：IMS，IXS 是为了让 MINICUBE 能够识别当前需要仿真的芯片型号，以便于设置仿真器对应的 RAM，ROM 空间大小。

2. 然后在定时中断服务程序（Interrupt Service Route）中添加自己的代码，让 LED 灯闪烁，即通过程序语句使 P32 和 P33 端口改变输出状态。

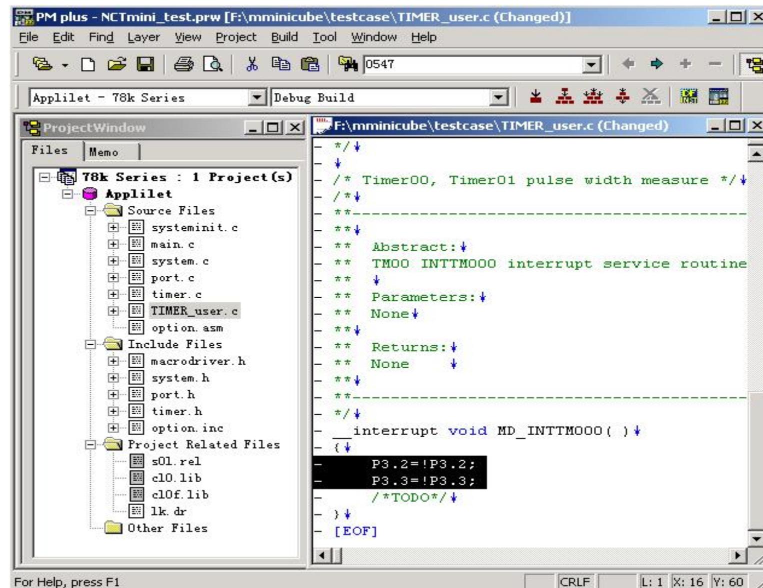


图 2-11 在 ISR 中添加自己的代码

在刚才使能的定时器 TM00 对应的中断服务程序中写两行代码：“P3.2=!P3.2；P3.3=!P3.3;”，即每经过 500 毫秒时间产生一次定时中断，中断过程将 P3.2 和 P3.3 的状态反转；该动作对应的表现就是诊断板上的红灯和绿灯亮灭。

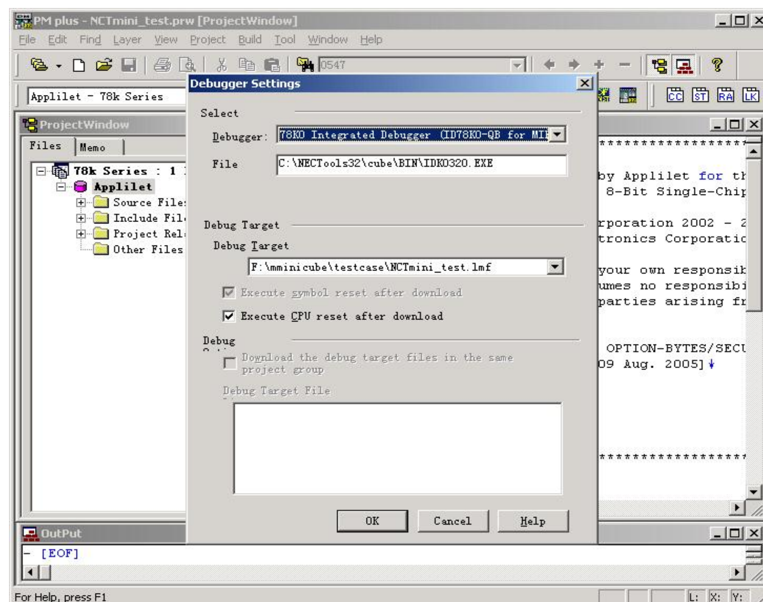


图 2-12 选择对应的调试器

第一次进入 PM+或者因为软件重装，如果没有看到菜单栏的调试器图标，那么就需要设置相应的调试器，点击“Tool→Debugger Settings”，选择“78K0 Integrated Debugger (ID78K0-QB for MINICUBE)”。如果没有找到此选项，请按照 1.4 节的步骤进行正确安装。

其他参数无需更改。所有链接和编译选项已经设置完成。用户直接点击“ReBuild”或者“Build and Debug”按钮。



图 2-13 Build 成功

如果用户手动编写程序，编译过程中若出现错误报警，需要反复修改，直到 Build 成功。

2.4　程序的运行和调试

2.4.1　启动 ID78K0-QB for MINICUBE

打开 ID78K0-QB for MINICUBE 程序，在“Chip”一栏中选择 78F0547_80。如果下拉框里面没有 78F0547_80 的选项，说明该芯片的设备文件没有被正确安装，请先安装设备文件，具体方法请参考 1.5 节。

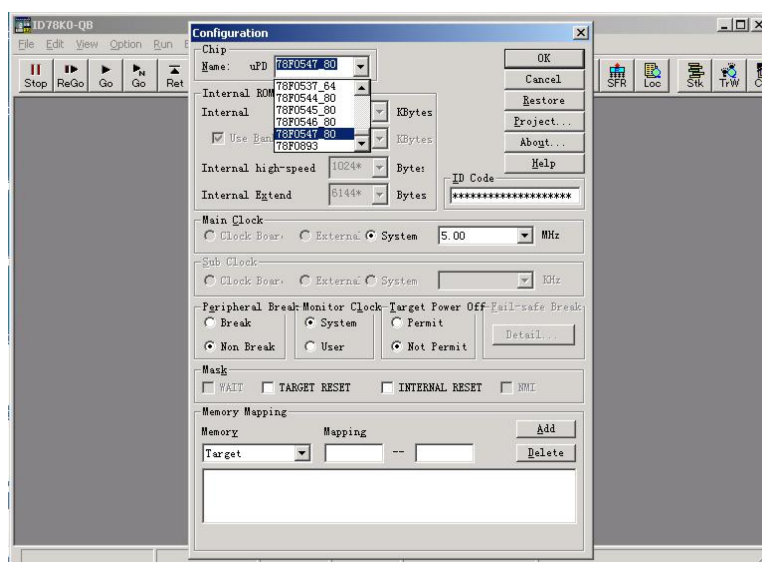


图 2-14　目标芯片的选择

如果出现如下画面的信息提示，原因是 USB 连线没有接好。请检查 USB 连线与主机的接口，以及 USB 连线与 MINICUBE 之间的连接情况。



图 2-15　ID 启动错误界面—USB 接线

- 如果是目标板的电源由 MINICUBE 提供，因为目标板没有连接好，那么出现的错误界面就如下所示。
- 如果目标板需要外接电源，那么目标板没有上电时，同样会出现这种错误：



图 2-16　ID 启动错误界面 -- 目标板未正确连接

对于以上两种错误，可以从 MINICUBE 的指示灯上迅速做出判断：

- 如果是图 2-15 的错误发生，那么 MINICUBE 的红色电源（POWER）灯处于熄灭状态，连接正确后，电源指示灯才会点亮。
- MINICUBE 中间的绿色目标（TARGET）灯是用于指示目标板的连接情况，如果是目标板的连接出现了问题或者目标板没有上电，那么中间的目标指示灯就不会亮。

正确的连接应该如下图所示：



图 2-17　MINICUBE 的指示灯

2.4.2　下载调试

用 ID78K0-QB for MINICUBE 下载工程文件“NCTmini_test.lmf”，可以看到以下下载界面：

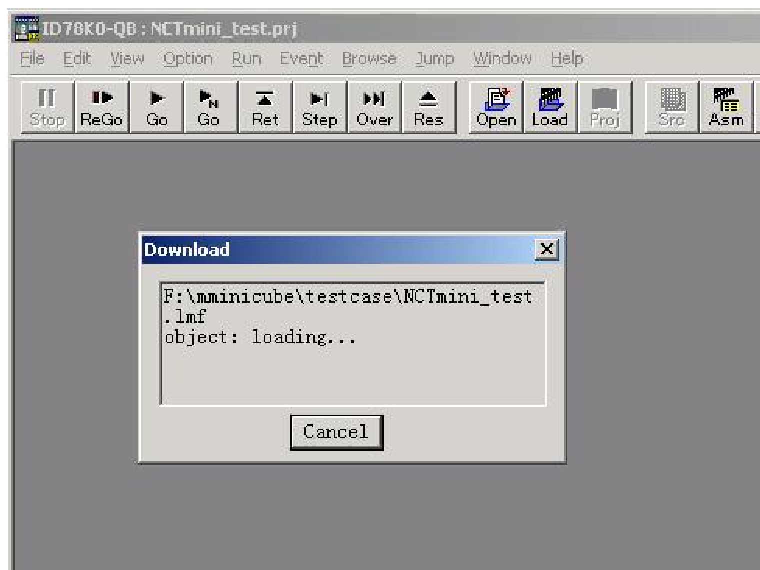


图 2-18　ID78K0-QB for MINICUBE 的下载界面

下载本程序大约需要 15 秒时间，程序下载时间取决于用户程序的大小。

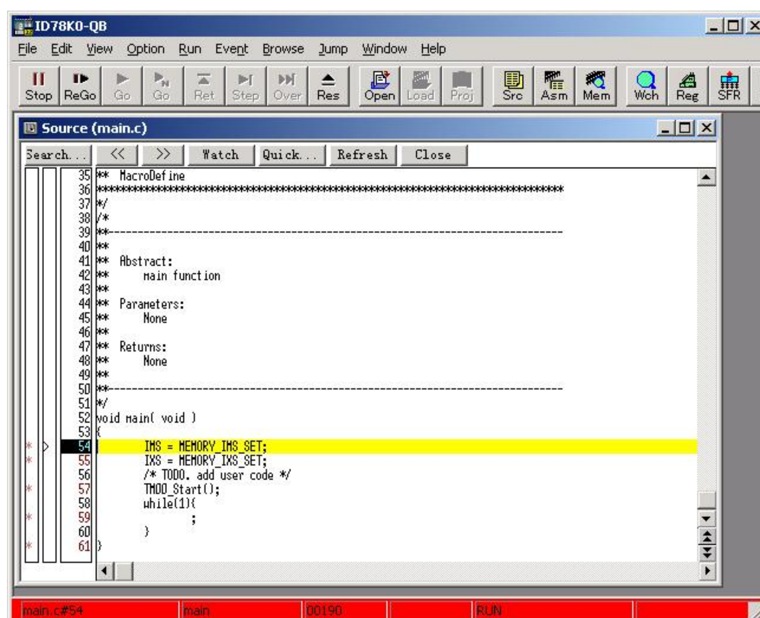


图 2-19　ID78K0-QB for MINICUBE 程序主界面

点击图 2-19 中的菜单栏按钮“REGO”，“GO”或者“GO-N”，程序开始运行，程序处于运行状态时，软件最底部的状态栏全都是红色的。

程序运行时的硬件效果图如下所示：



图 2-20 诊断板上的 LED 灯开始闪烁

如果工作不正常，请检查硬件连接和跳线设置情况。

第 3 章 NEC Tools 开发环境简介

本章将简要介绍手册中涉及到的 NEC Tools 开发工具。

3.1 概述

NEC Tools 开发环境包含了 NEC 系列微处理器（MCU）开发所需要的多种工具，如驱动代码生成工具、编译工具、链接工具、调试工具等。

本手册将在后面的章节详细介绍用于 78K0 系列 MCU 的 NEC 开发工具，其中包括驱动代码生成工具 Applilet、集成编译环境 PM+、集成调试环境 ID78K0-QB for MINICUBE。

3.2 C 编译器 CC78Kx

CC78Kx 已经集成到 PM+平台中，用户可以使用 PM+编译工程，也可以抛开 PM+，直接使用 CC78Kx 的命令行方式编译工程。

CC78Kx 它读入 C 源文件，编译生成二进制文件，或者是汇编文件。

CC78Kx 的输入文件包括：

file.c　　C 源文件

CC78Kx 的输出文件包括：

file.asm　　汇编源文件

模块文件

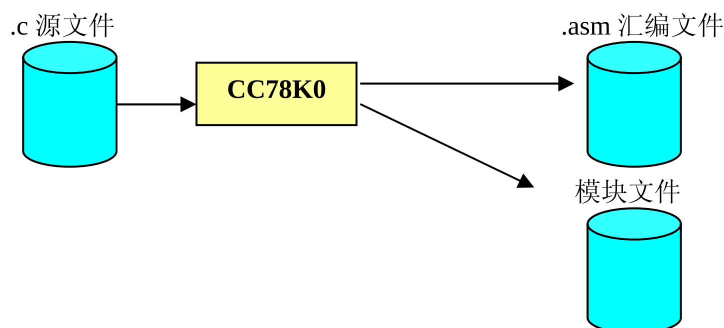


图 3-1 CC78Kx 工作示意图

3.3 汇编器 RA78Kx

RA78Kx 已经集合到 PM+平台中，用户可以使用 PM+编译工程，也可以抛开 PM+，直接使用 RA78Kx 的命令行方式编译工程。

RA78Kx 读入汇编源文件，编译成目标文件，目标文件中的段和变量待定位，直到它们通过链接工具链接之后，才分配确定的内存地址。

RA78Kx 的输入文件包括：

file.asm　　汇编文件

AS78Kx 的输出文件包括：

file.rel　　目标文件

file.prn　　列表文件

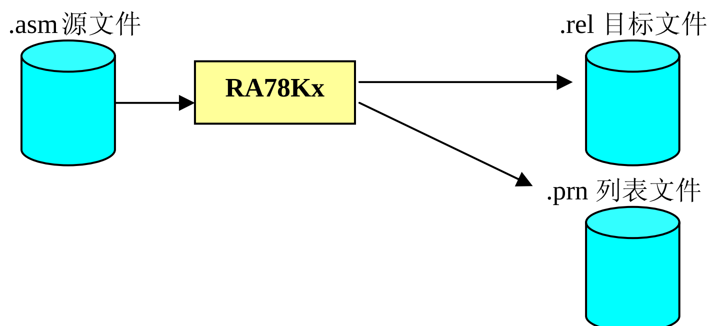


图 3-2　RA78Kx 工作示意图

3.4　链接器 LK78Kx

LK78Kx 已经集合到 PM+平台中，用户可以使用 PM+链接工程，也可以抛开 PM+，直接使用 LK78Kx 的命令行方式链接工程。

一般情况下，用户程序由多个源文件组成。当用 CC78Kx 和 RA78Kx 来处理源文件以后，已经生成了多个目标文件（.rel），目标文件中的段和寄存器地址未定。此时用 LK78Kx 链接它们，生成一个可执行的模块文件（.lmf）。

除了源文件，LK78Kx 在链接过程中还需要读入段链接文件（Directive file，扩展名.dr），以获取段地址和大小信息；读入设备文件（Device file，扩展名.78k），以获得特殊寄存器和芯片内存信息；读入库文件（Library file，扩展名.lib），以获得库函数代码。

LK78Kx 的输入文件包括：

file.rel　　目标文件
file.dr　　段链接文件
file.78k　　设备文件
file.lib　　库文件

LK78Kx 的输出文件包括：

file.lmf　　可执行模块文件

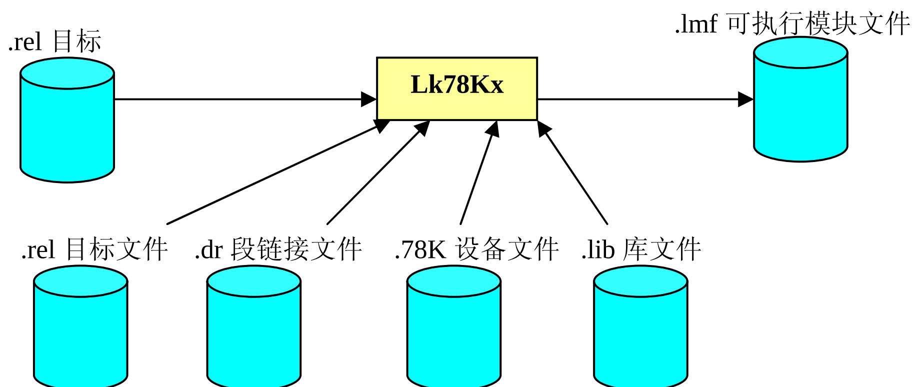


图 3-3　LK78Kx 工作示意图

3.5 目标文件转化器 OC78Kx

用户可以调用 OC78Kx 来转化可执行模块文件（.lmf）格式为可执行文本文件（.hex），并且可以输出符号表文件（.sym）。

OC78Kx 支持以下 hex 标准格式：Intel 格式。

OC78Kx 的输入文件包括：

file.lmf　　可执行二进制文件

OC78Kx 的输出文件包括：

file.hex　　可执行文本文件

file.sym　　符号表文件

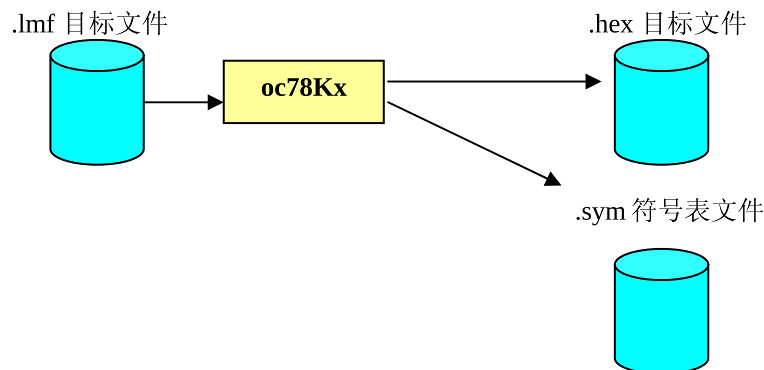


图 3-4 OC78Kx 工作示意图

3.6 库生成器 LB78Kx

LB78Kx 已经集合到 PM+平台中，用户也可以抛开 PM+，直接调用 LB78Kx 来将多个二进制文件（.rel）链接为库文件。

LB78Kx 的输入文件包括：

file.rel　　目标文件

LB78Kx 的输出文件包括：

file.lib　　库文件

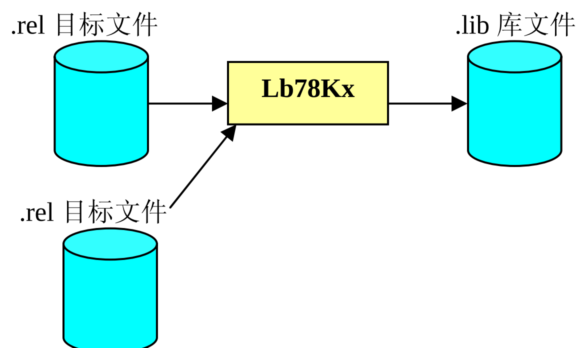


图 3-5 LB78Kx 工作示意图

第 4 章　驱动代码生成工具 Applilet

本章介绍 Applilet 驱动代码生成工具，通过图形化的界面，可以方便的配置你所需要的模块功能，根据图形界面的配置参数，帮助用户生成程序驱动框架，只需要做少量修改，就能得到高质量的 C 代码或者汇编代码。用户不需要阅读相关的手册阅去了解每一个寄存器如何设置。

范例程序中使用的是 Applilet for 78K0KX2，专用于 NEC 78K0 KX2 系列 MCU。

4.1　启动 Applilet

点击“开始→程序→ NEC Tools→Applilet”图标 ，将会弹出如下界面：

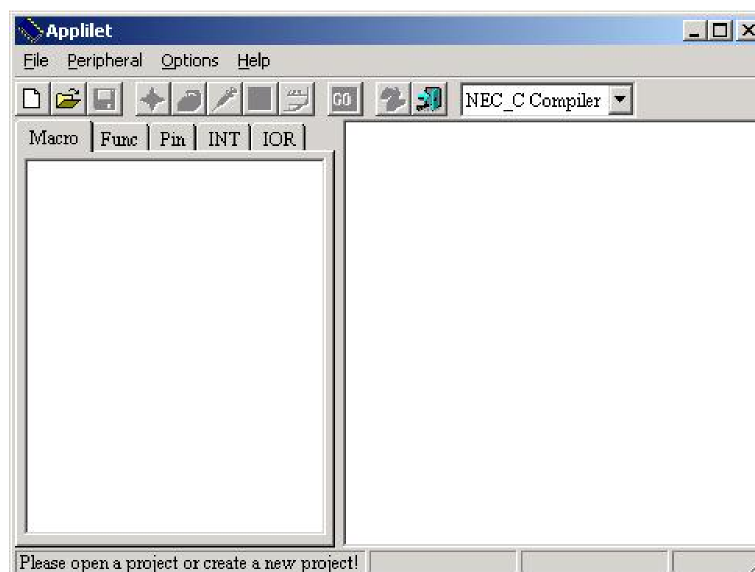


图 4-1　Applilet 主界面

4.2　功能界面

点击“File→New”就看到了 Marco 的图形化界面，如图 2-2 所示，其中 Macro 模块根据 MCU 的种类不同而有所差异。

用户可以在“Func”栏观察代码的函数框架，了解图形界面上参数所对应的寄存器详细配置情况，如下图所示：

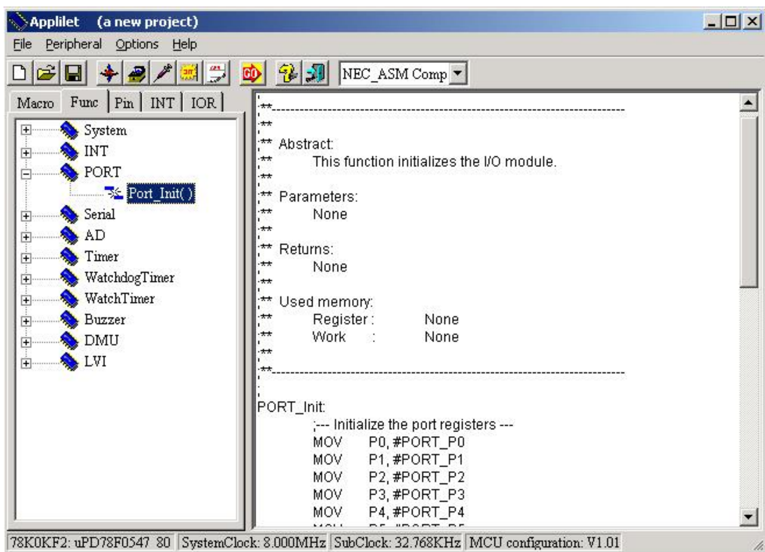


图 4-2　Applilet 的 Func 界面

可以通过 Applilet 观察芯片上的所有引脚资源，以及已经被用户程序占用的资源，如下图所示：

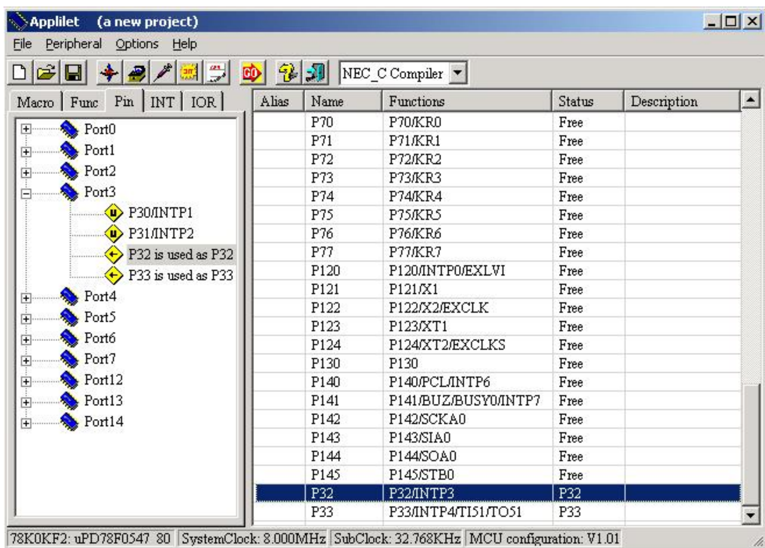


图 4-3　Applilet 的 Pin 界面

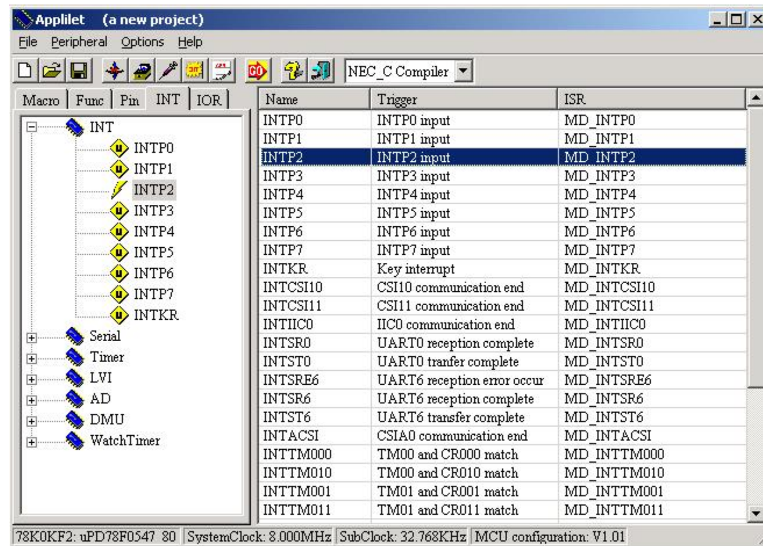


图 4-4　Applilet 的 INT 界面

上图右侧的 ISR 是用户设定的功能在生成的代码中的函数名称。

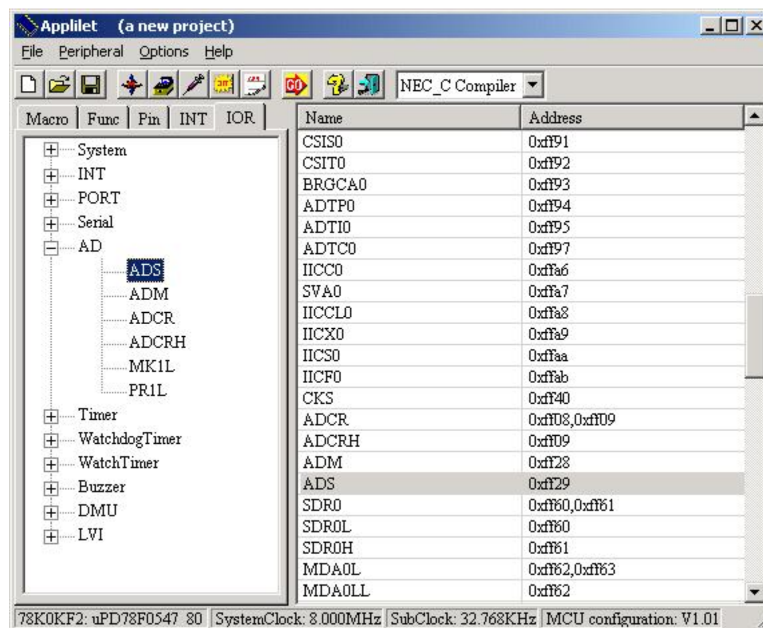


图 4-5　Applilet 的 IOR 界面

上图列出了每一个模块所需要的所有寄存器名称和在存储器中映射的地址，方便用户查询。

4.3　模块介绍

μPD78F0547DGC 芯片内部包括了 11 个模块，分别是 System，INT，PORT，Serial，AD，Timer，WatchdogTimer，WatchTimer，Buzzer，DMU，LVI。

表 4-1　μPD78F0547DGC 模块介绍

模块	功能简介
System	系统基本设置
INT	各种中断相关的设置

PORT	外部输入/输出终端设置
Serial	串口通讯的设置
AD	AD 转换的设置
Timer	时钟的设置
Watchdog Timer	看门狗定时器的设置
WatchTimer	监视定时器的设置
Buzzer	蜂鸣器的设置
DMU	乘法除法模式设置
LVI	低压检测设置

各个模块的具体功能和设置请参阅《78K0/KF2 8-Bit Single-Chip Microcontrollers Preliminary User’s Manual》（U17397E）。

第 5 章　集成开发环境

本章介绍如何使用 PM+工具。

5.1　使用 PM+编译

5.1.1　启动 PM+

点击“开始→程序→ NEC Tools→PM+”图标，将会弹出如下界面。如果不是第一次执行，PM+会自动加载最后一次执行的 Workspace。

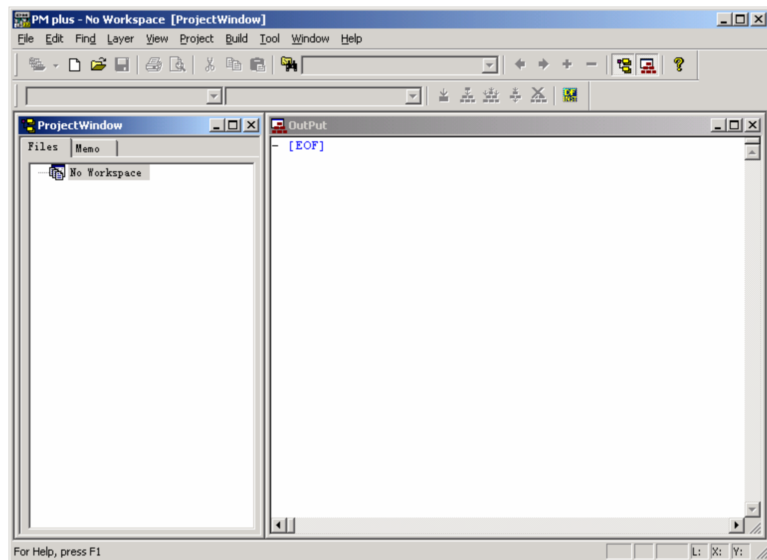


图 5-1　启动 PM+

5.1.2　工程术语

Project：

Project 是被 PM+管理的一个单元，指在 PM+环境下开发的一个应用系统。PM+将每一个应用系统用到的源文件、目标设备名称、模块的工具选项以及编辑器或调试器等信息保存到一个项目文件（*.prj）里。编译或调试都是在项目单元里进行的，因此，要编译或调试的项目必须被设置为“当前项目（active project）”，可以通过选择“Project”菜单中的“Select Active Project…”菜单项进行设置。

Workspace：

Workspace 是一个管理多个项目（project）文件的单元，PM+把这些项目文件的文件名保存到一个 workspace 文件（*.prw）中。

Project Group：

多个已注册的项目可以和相关的项目组成一个组，这个组就是 Project Group。注意一个项目组中必须使用相同的设备文件。

IDL file：

IDL file 是一个保存 PM+层次关系的文件。

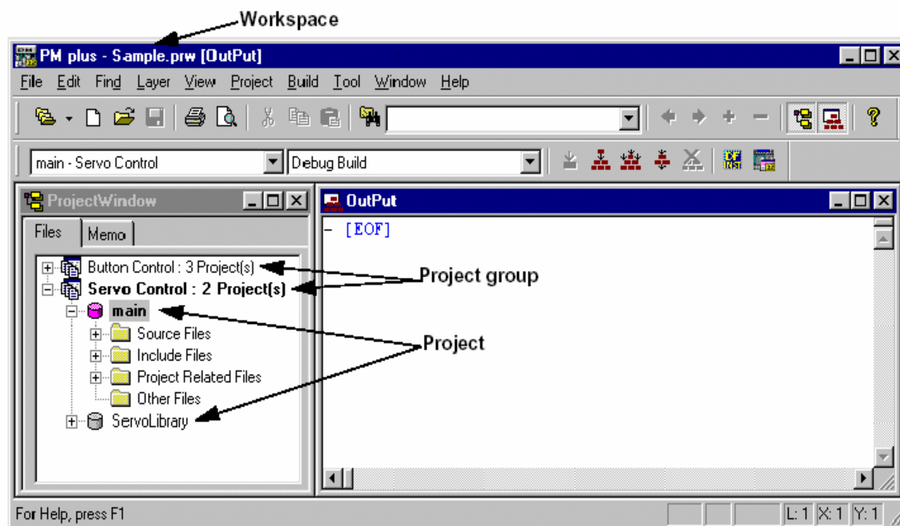


图 5-2 PM+操作界面

5.1.3 新建 Workspace

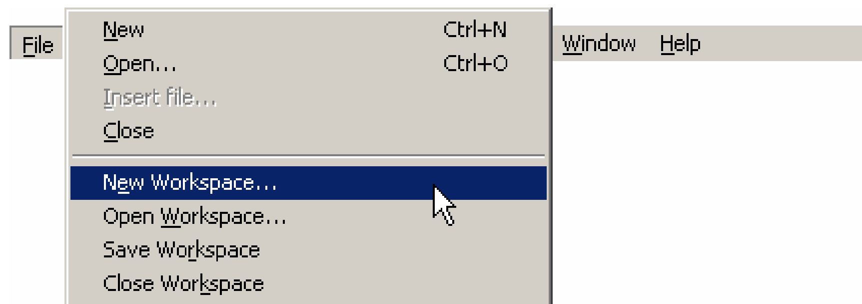


图 5-3 新建 Workspace

要使用 PM+管理项目，之前必须先建立一个 Workspace。选择“File”菜单中的“New Workspace…”，将打开新建对话框。

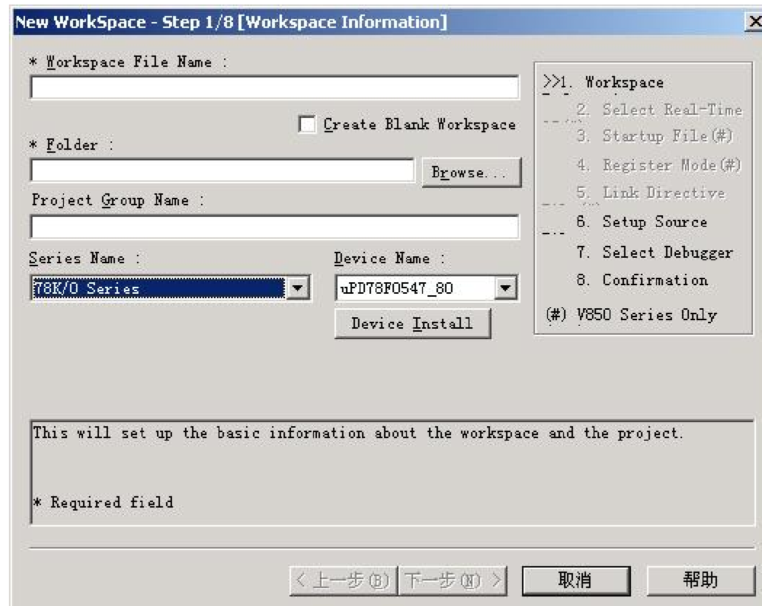


图 5-4　New WorkSpace – Step 1/8 对话框

Step1/8 对话框中设置如下内容：

Workspace File Name　新建的 Workspace 名称。这个名称将显示在标题栏上。

Folder　所有工程文件的存放目录。

Project Group Name　新加入的 Project Group 名字。

Series Name　Project 使用的设备系列名称。下拉菜单提供已安装的所有设备系列。

Device Name　Project 对应的设备名称。下拉菜单提供已安装的所有设备名称。

填好设置，点击“下一步”按钮，出现创建对话框 Step6/8。

注：Step2/8~step5/8 是专门为 V850 系列芯片提供的功能，分别为选择操作系统，选择启动文件，寄存器模式，程序代码和数据的存放位置文件。

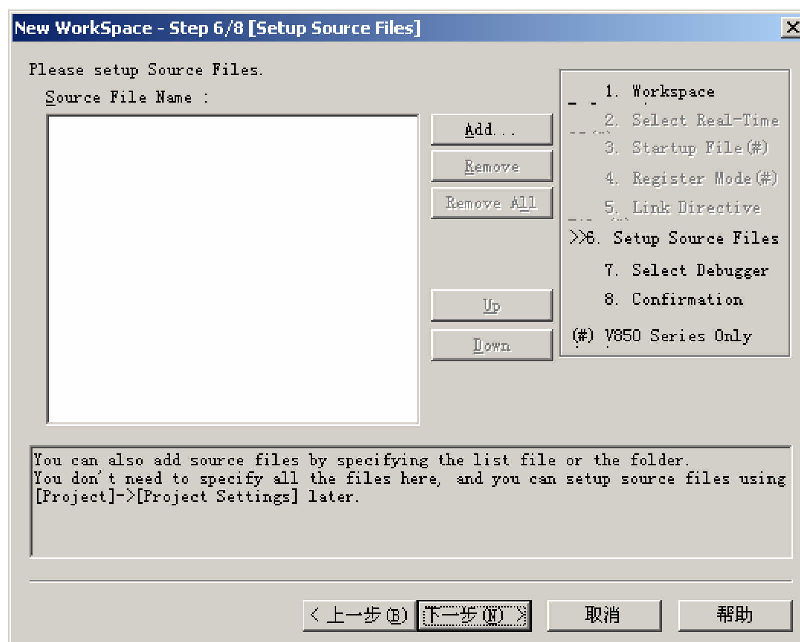


图 5-5　New WorkSpace – Step 6/8 对话框

在 Step6/8 中添加源文件。点击“Add”按钮来添加源文件，可以是 C 文件或汇编文件。

点击对话框“下一步”按钮，出现创建对话框 Step7/8。

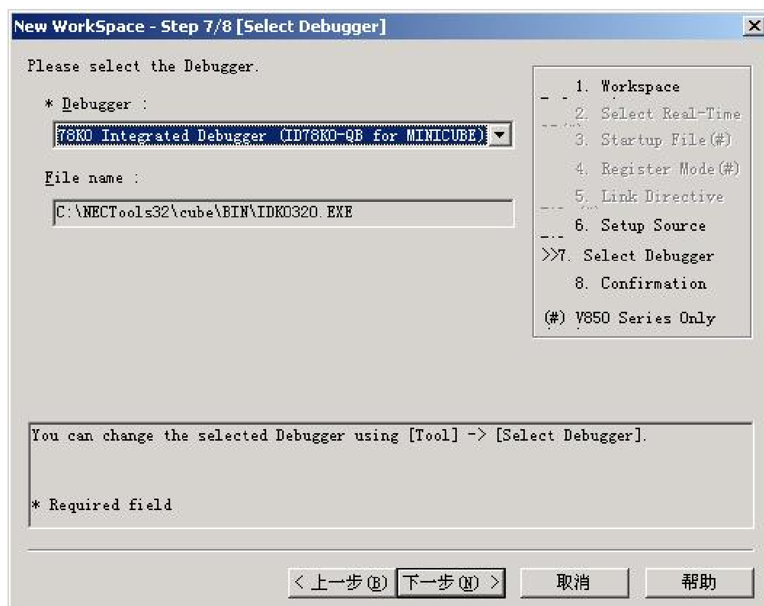


图 5-6　New WorkSpace – Step 7/8 对话框

对话框 Step7/8 中选择调试器，下拉菜单提供了已安装的所有调试器。

点击对话框“下一步”按钮，出现创建对话框 Step8/8。

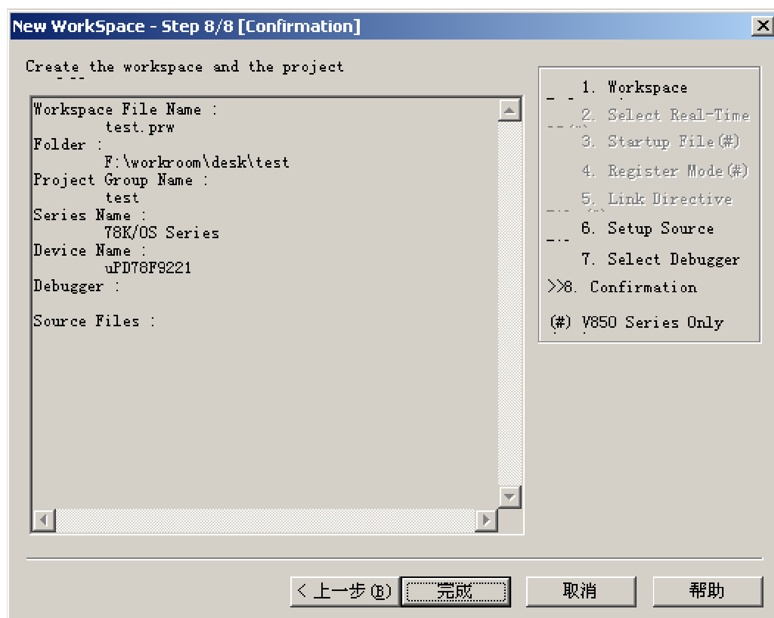


图 5-7　New WorkSpace – Step 8/8 对话框

对话框 Step8/8 是创建完成信息对话框。

点击“完成”，一个全新的 WorkSpace 就建立好了。

5.1.4 打开 Workspace

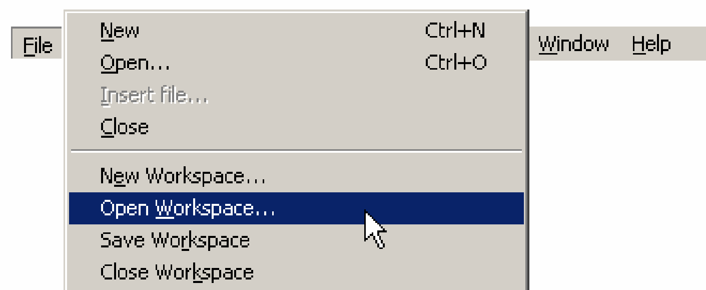


图 5-8 打开 Workspace

如果要加载已有工程，选择“File”菜单中的“Open Workspace”，打开 Open Workspace 对话框：

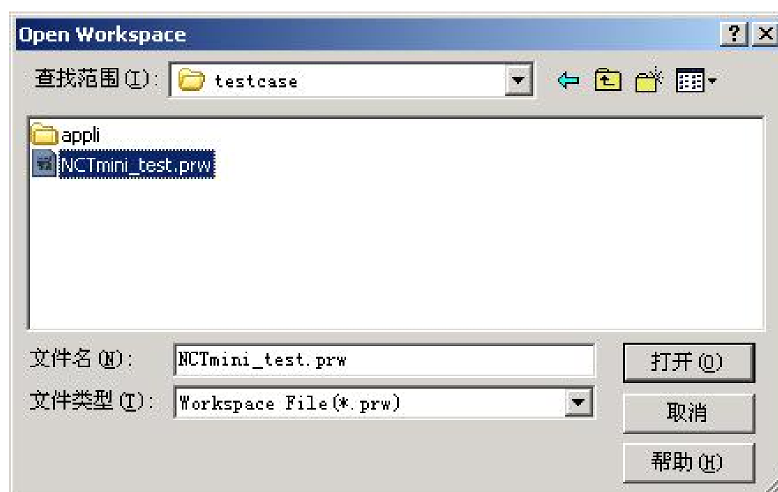


图 5-9 Open Workspace 对话框

选择 Workspace 文件（*.prw），点击打开按钮，就可以读取相应的 Workspace。

5.1.5 Build 工程

根据设置的不同，Build 一个工程将产生二进制代码文件（*.lmf），或是库文件（*.lib）。

Build 工程只需点击 Build 按钮，或是从菜单中选择“Build→Build”。

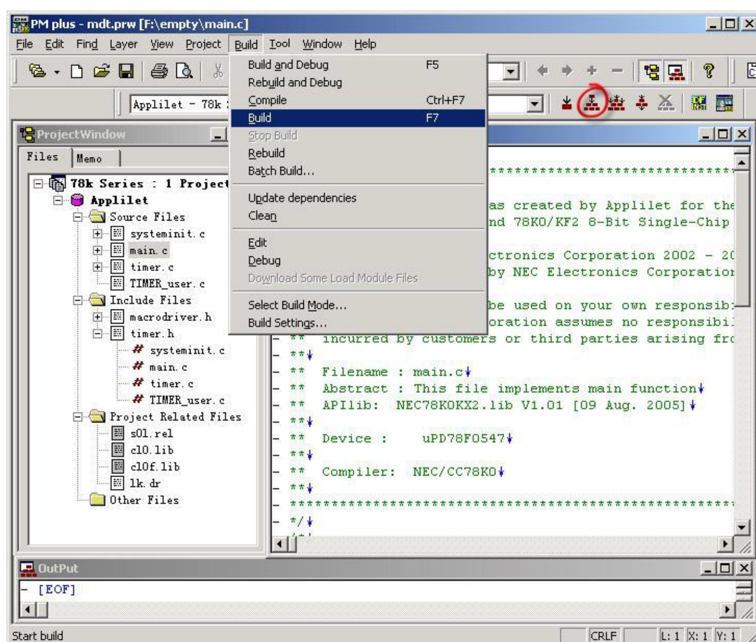


图 5-10　Build 工程

如果编译链接都正确，那么会出现消息框。



图 5-11　Build 成功

这时候就可以进行下一步的调试了。

5.2　选项设置

PM+为用户提供了丰富的高级设置功能。

5.2.1　编译选项

选择“Tool”菜单中的“Compiler Options…”，可以打开 Compiler Options 对话框，提供了编译、优化、预处理等的配置选项。

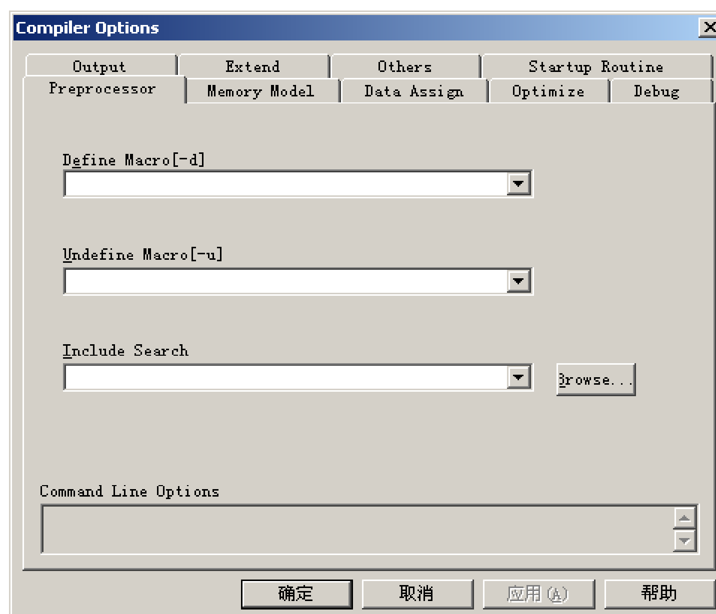


图 5-12　Compiler Options 对话框

5.2.2　汇编选项

选择“Tool”菜单中的“Assembler Options…”，打开 Assembler Options 对话框，提供了汇编语言的配置选项。

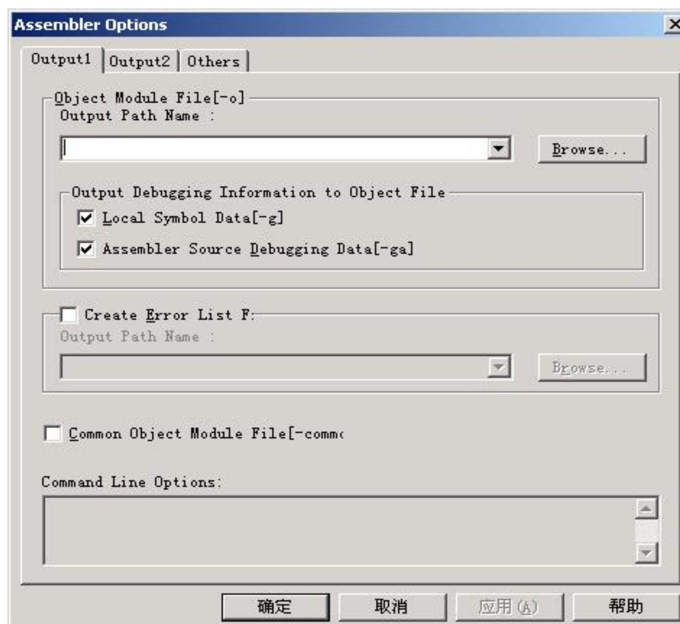


图 5-13　Assembler Options 对话框

5.2.3　链接选项

选择“Tool”菜单中的“Linker Options…”，打开 Linker Options 对话框，提供了段链接文件、用户自定义库等的配置选项。

1. 如果需要使用 MINICUBE 调试，那么要注意在此处打开 OCD 功能，否则无法进入调试器。选中图 5-14 的 On-Chip Debug 选项就打开了 OCD 功能。

2. 如果也选中了“Security ID”，就可以设置密码。即使用户使用默认 Security ID，同样也建议用户激活该选项（如图 5-14 所示），以确保用户程序不会覆盖占用这几个字节的内容。
3. 在此处如果没有打开 OCD，用于将不能使用 ID78K0 Debugger 下载和调试程序。但是打开了 OCD，也不能保证一定可以进入调试器。源文件中必须包含“option.asm”文件，使能“Linker option”的 OCD 选项和“option.asm”文件两者缺一不可，这样才能保证进入调试器。
4. “option.asm”文件的内容如下：

```
OPT DSEG AT 0080H
                    ;option byte setting
OPTION: DB 00H      ; stop watchdog timer, stopable internal low-speed oscillator by software
DB 00H              ; 1.59V POC mode
DB 00H              ; reserve area
DB 00H              ; reserve area
DB 03H              ; On-chip debug operation is enabled.
                    ; Erases data of the flash memory when authentication of security ID fails.
END
```

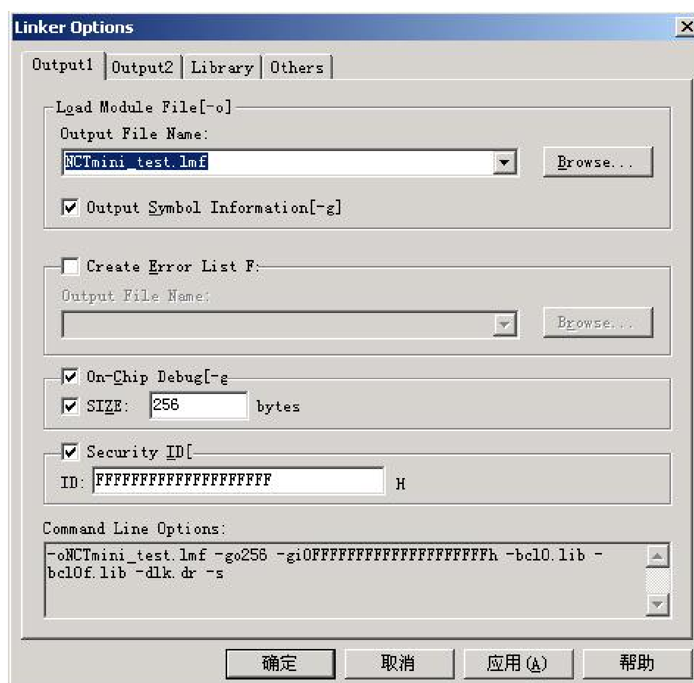


图 5-14　Linker Options 对话框

5.2.4　编译模式

需要注意的是 Build 模式分为“Debug Build”和“Release Build”。一般需要调试的情况我们都选择“Debug Build”模式，使用这种模式生成的文件会包含一些调试信息，方便我们使用 ID78K0-QB for MINICUBE 进行调试观察。

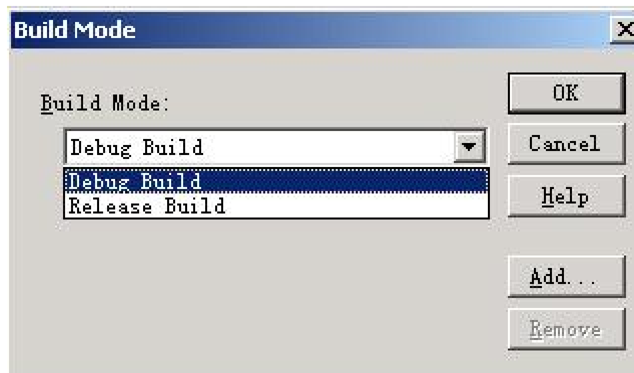


图 5-15　Build 模式的选择

Release 模式生成的文件在 ID78K0 中无法调试，会发生下图的错误信息：



图 5-16　警告信息

其中每个选项的参数具体如何设置，以及设置的内容会引起的后果，对程序会产生什么样的影响，这些问题的解决敬请查阅《PM plus Ver5.20 User's Manual》（U16934E），这个文件在安装 PM+软件的时候会自动加载，打开方式为“开始→程序→NEC tools32→78K0 tool documents→PM plus→UM V5.20 PM plus”。

第 6 章　NCT-MINICUBE78K0 介绍

本章介绍 NCT-MINICUBE78K0 硬件环境的建立。

6.1　NCT-MINICUBE78K0 的基本特性

6.1.1　产品特点

1. 可用于调试 NEC 78K0 全系列 MCU，但是目标设备必须内置 OCD 单元（产品编号 xxxxD）
2. 可以对已经安装在目标板上的 MCU 进行调试
3. 占用 X1X2 引脚进行通讯，也可以通过其他端口通信
4. 可以使用 Flash 自编程功能将程序下载到 Flash 存储器中
5. 不需要外部电源（USB 供电）
6. 兼容 USB2.0
7. 使用自测程序和诊断板来维护
8. 外形小巧（84.5*56.5*26.1mm，重约 60 克）
9. 具有伪实时 RAM 监测和动态存储器修改功能

6.1.2　包装配件表

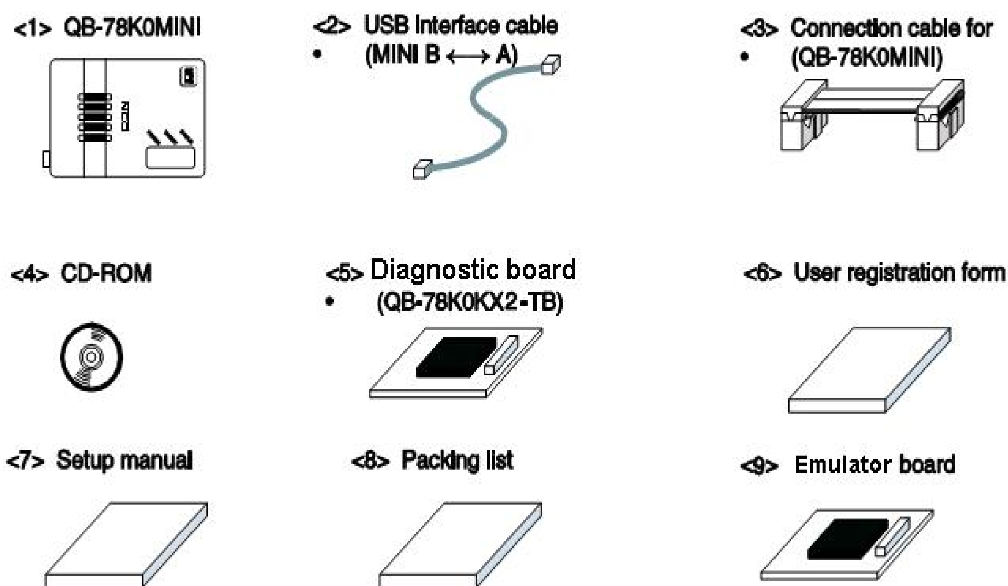


图6-1　MINICUBE 包装配件图

其中包括 9 项内容：

（1）NCT-MINICUBE78K0

（2）USB 接口电缆（长度：2 m）

（3）NCT-MINICUBE78K0 连接线（长度：20 cm）

（4）安装光盘和资料光盘各一份

（5）诊断板（QB-78K0KX2）

（6）用户登记表

（7）安装手册

（8）装箱单

（9）仿真板

6.1.3　MINICUBE 外观

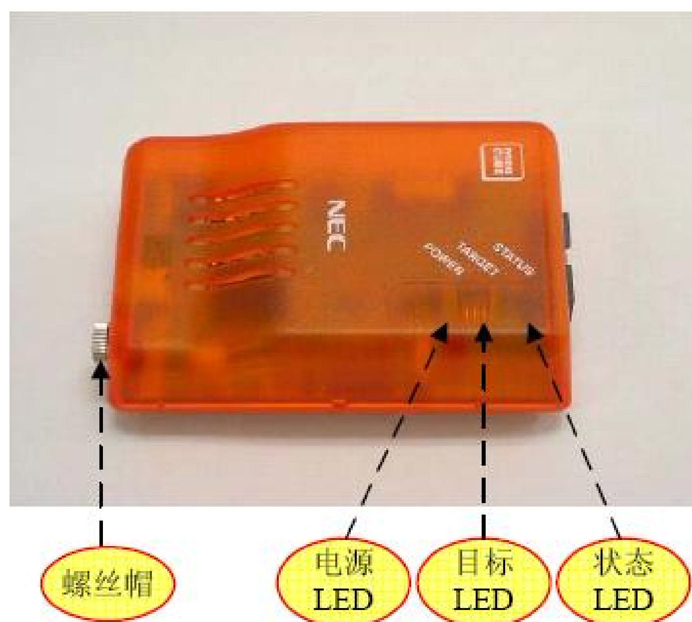


图 6-2　MINICUBE 主视图



图 6-3　MINICUBE 左视图



图 6-4　MINICUBE 右视图

6.1.4　MINICUBE 内部电路板

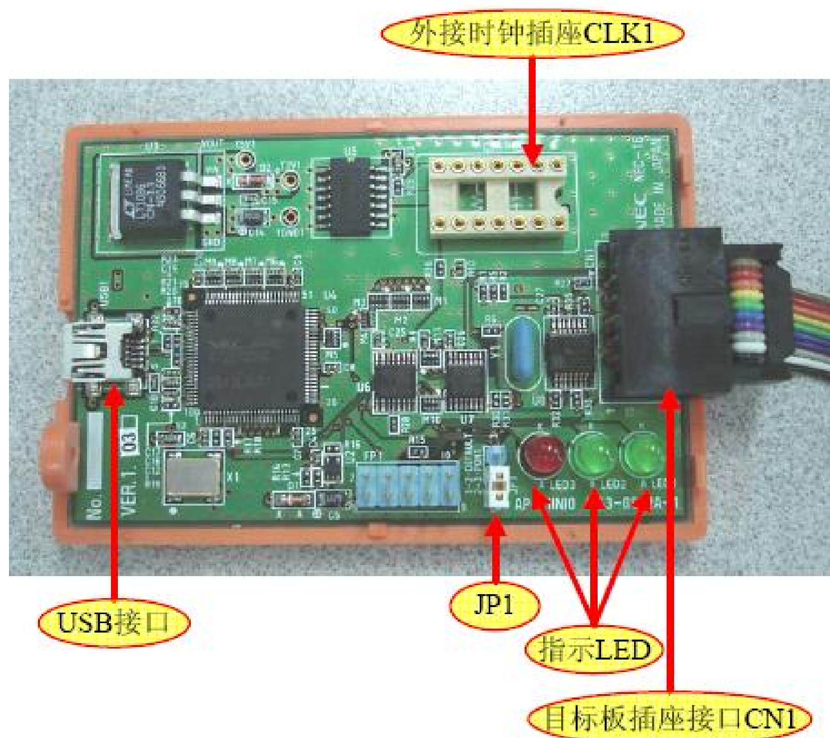


图 6-5　MINICUBE 内部构造

要在上图中的外接时钟插座上安装振荡器，敬请参阅《NCT-MINICUBE78K0 On-Chip Debug Emulator User’s Manual》（U17029E）第三章的内容。

6.1.5　NCT-MINICUBE78K0 同相关设备的连接

警告：在连接任何相关设备之前，确保在主机上已经正确安装了 ID78K0-QB 调试软件，MINICUBE 的 USB 驱动，OCD Checker 软件和相关的设备文件。

接下来描述如何将 NCT-MINICUBE78K0 同其他相关设备连接的过程。

1. USB接口电缆的连接（NCT-MINICUBE78K0端）
   将 USB 连接电缆的 MINI-B 端同 NCT-MINICUBE78K0 的 USB 连接器相连。
2. USB接口电缆的连接（主机端）
   将 USB 连接器的 A 连接端接入主机的 USB 端口。
3. NCT-MINICUBE78K0与目标板的连接方式请参阅6.2节。
4. 给MINICUBE上电
   （1）当使用诊断板时
   USB 电缆连接好后，如果诊断板需要独立供电，必须对其加电。正常情况下 NCT-MINICUBE78K0 的绿色目标指示灯（TARGET）和红色电源指示灯（POWER）都应该是点亮的。
   （2）当使用目标系统时
   当 USB 电缆连接到主机后，只有 NCT-MINICUBE78K0 的红色电源指示灯（POWER）是亮的。当打开目标系统的电源后，NCT-MINICUBE78K0 的绿色目标指示灯（TARGET）也应该是亮的。
5. 启动ID78K0-QB
   确定 NCT-MINICUBE78K0 连接正确，打开诊断板或目标系统的电源，启动 ID78K0-QB for MINICUBE。

6.1.6　NCT-MINICUBE78K0 同相关设备的断开

以下步骤描述了断开NCT-MINICUBE78K0同相关设备连接的过程。

1. 关闭ID78K0-QB。
2. 关闭目标系统的电源（如果使用的是目标系统）。
3. 移去NCT-MINICUBE78K0和主机间的USB接口电缆。
4. 移去NCT-MINICUBE78K0同目标系统或者诊断板之间的NCT-MINICUBE78K0连接电缆。

上面是推荐的断开步骤，不按照这个顺序退出可能会对硬件造成潜在影响。

6.1.7　NCT-MINICUBE78K0 连接器针脚设置

NCT-MINICUBE78K0 的连接接口描述如下：

表 6-2　NCT-MINICUBE78K0 连接器针脚设置

针脚号	名称	输入/输出	备注
1	RESET_IN	输入	目标系统复位输入
2	RESET_OUT	输出	目标系统 CPU 复位输出
3	FLMD0	输出	
4	VDD_IN	输入	目标系统电源供给探测
5	X2	输入/输出	N-CH 连接（针对 TXD 和 RXD）

6	GND	—	
7	X1	输出	时钟供给
8	GND	—	
9	RESERVED	—	
10	5V_CHK	—	诊断板电源供给

注：输入输出的方向是针对 NCT-MINICUBE78K0 而言

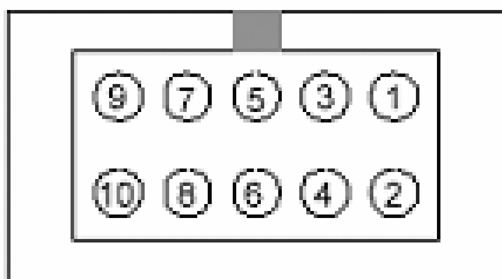


图 6-6　MINICUBE 连接器针脚布局

6.1.8　NCT-MINICUBE78K0 和目标 MCU 的连接电路

NCT-MINICUBE78K0 和目标 MCU 之间的连接有以下三种电路方式：

1. 不连接 NCT-MINICUBE78K0 时 X1，X2 的电路连接图例

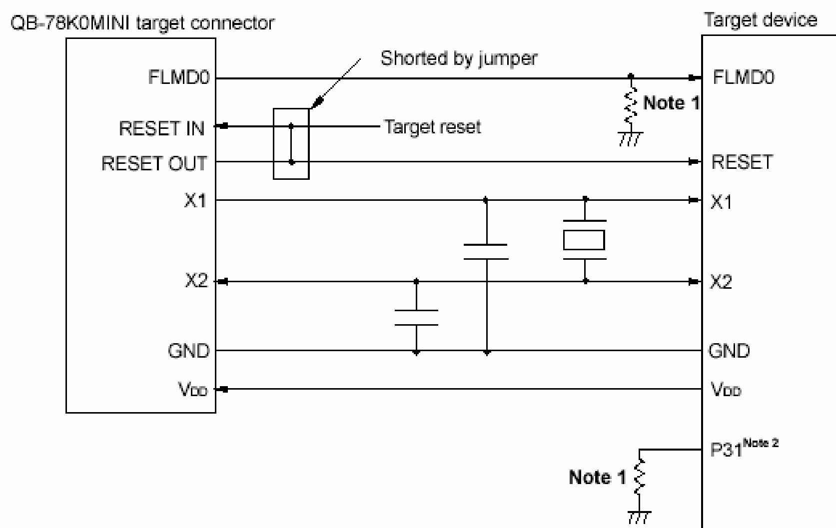


图 6-7　连接电路图例 1

注：A．下拉电阻值的描述，参见目标设备的用户手册，一般 470 欧姆或更高。

B．当使用 78K0/KF1+时

2. 通过 X1、X2 连接 NCT-MINICUBE78K0 时

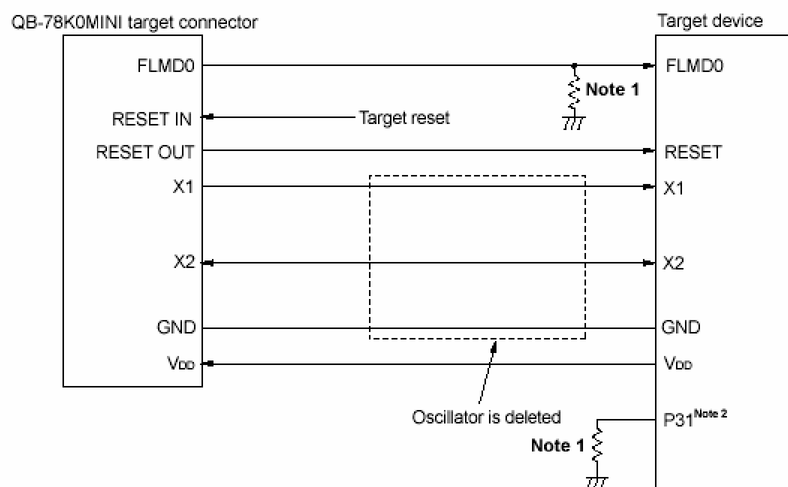


图 6-8　连接电路图例 2

注：A．下拉电阻值的描述，参见目标设备的用户手册，一般 470 欧姆或更高。

B．当使用 78K0/KF1+时

3.　通过端口 AB 连接 NCT-MINICUBE78K0 时

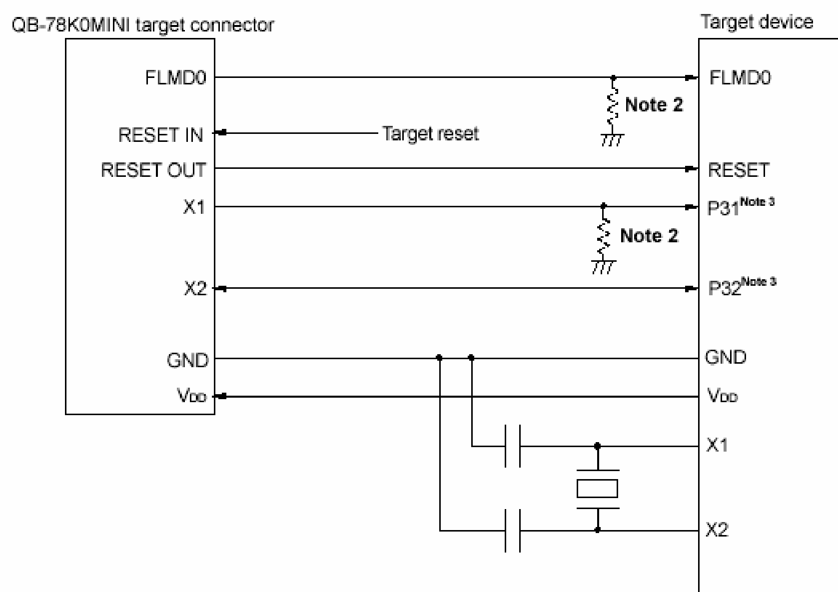


图 6-9　连接电路图例 3

注：A．端口 A 和端口 B 管脚根据目标设备的不同而有所差异。

B．下拉电阻值的描述，参见目标设备的用户手册，一般 470 欧姆或更高。

C．当使用 78K0/KF1+时，P32 是端口 B，产品型号不同，引脚端口也不同。

6.2 NCT-MINICUBE78K0 的系统搭建方式

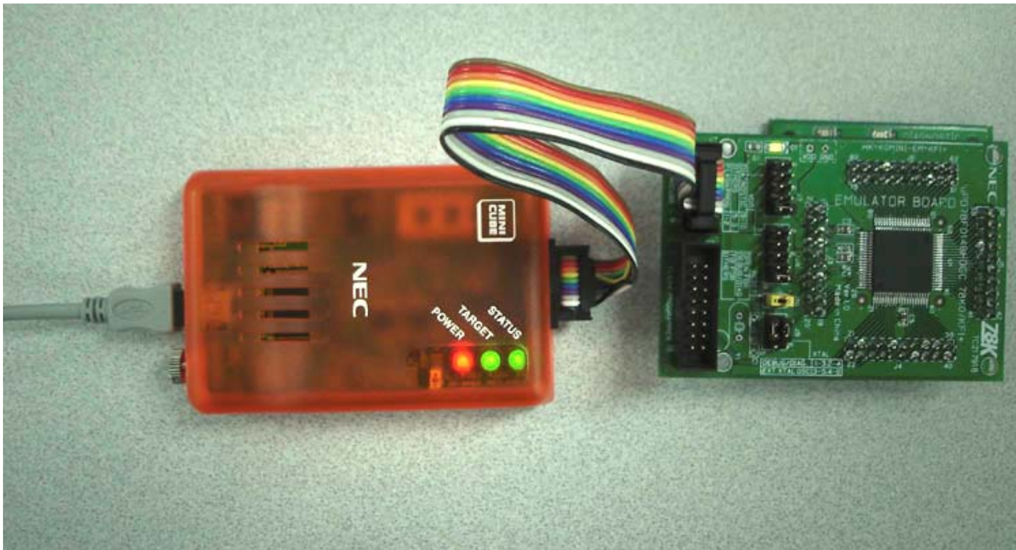


图 6-10 MINICUBE 的系统连接图

NCT MINICUBE向外的连接方式一共有三种：

通过仿真板连接诊断板；

通过排线连接到目标板；

通过仿真板以ICE方式连接到目标板。

三种方式都要求使用USB线连接主机和MINICUBE。需要使用 10 针排线连接MINICUBE到仿真板或目标板的时候，注意排线接口上有用于识别插口方向的卡口。

6.2.1 通过仿真板连接诊断板

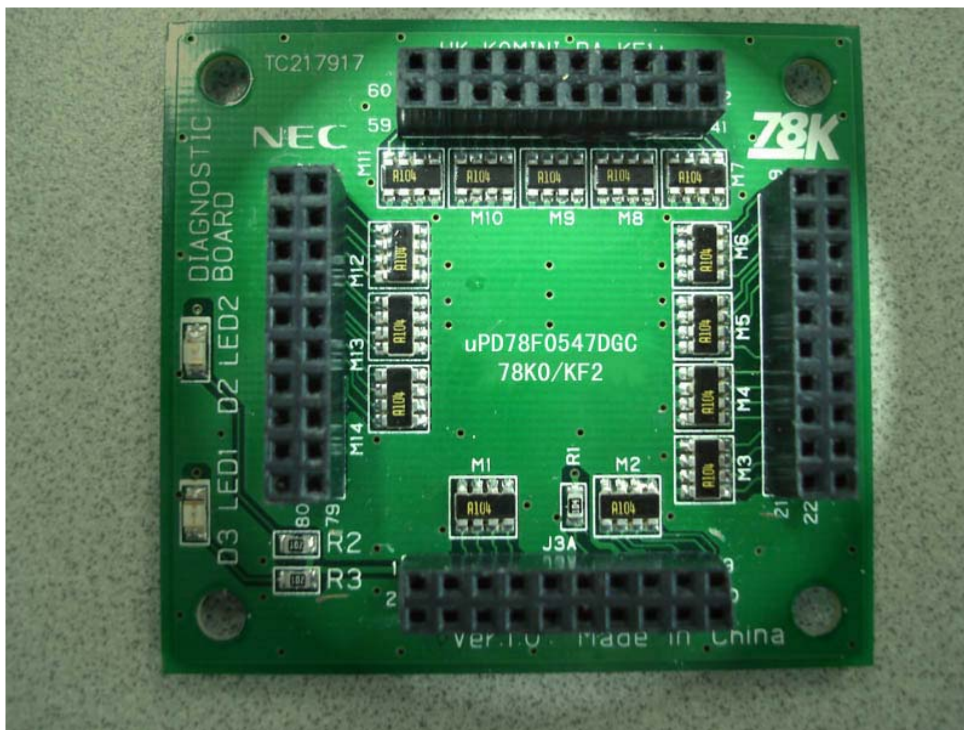


图 6-11 NCT-MINICUBE78K0/KF2 的诊断板

新版诊断板很简单整洁，没有芯片，只有四排 20 针的插座和一红一绿两个LED灯。这两个LED灯主要用于MINICUBE的自检维护。

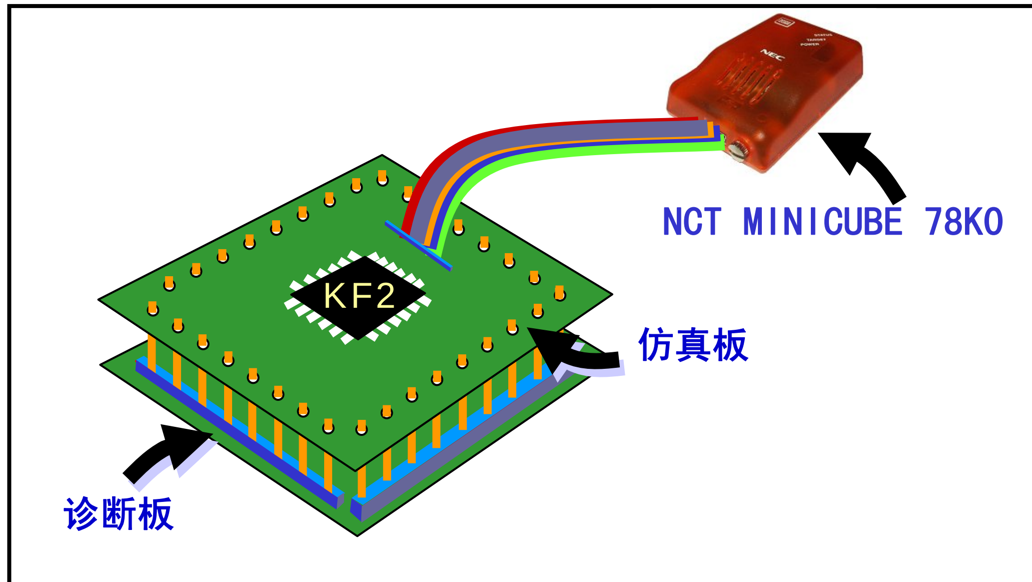


图 6-12　MINICUBE 通过仿真板连接诊断板的示意图

其中仿真板和诊断板在NCT MINICUBE的包装盒里附带的。

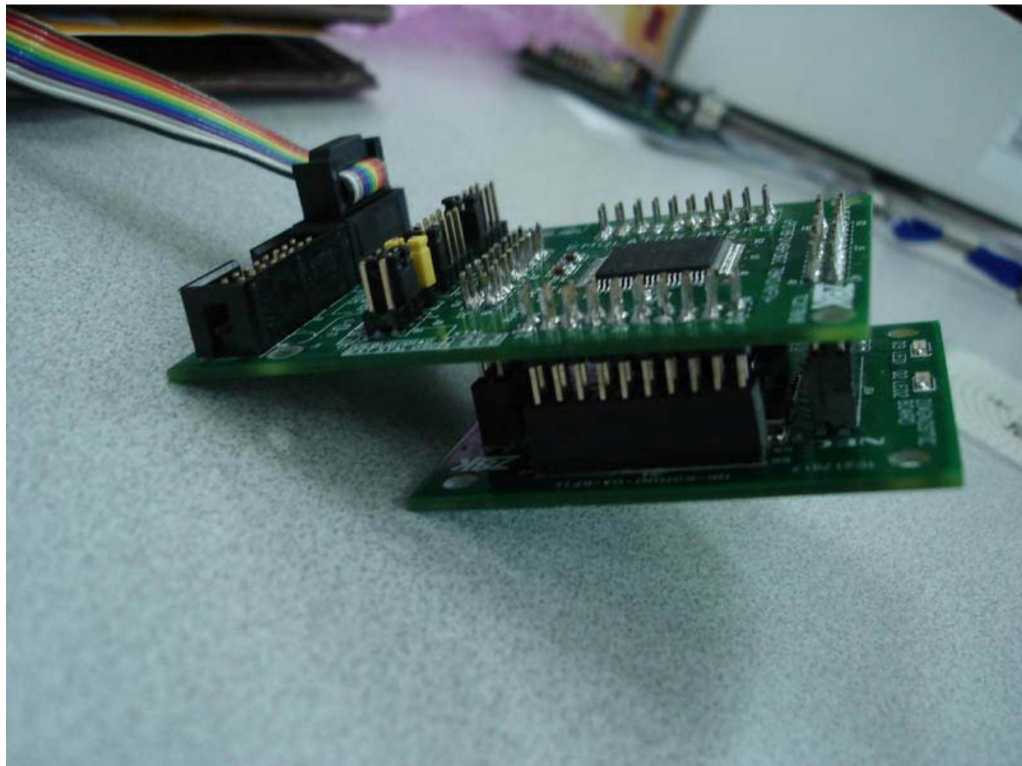


图 6-13　MINICUBE 通过仿真板连接诊断板的实物图

通过仿真板连接诊断板时，仿真板上的跳线设置如下图：

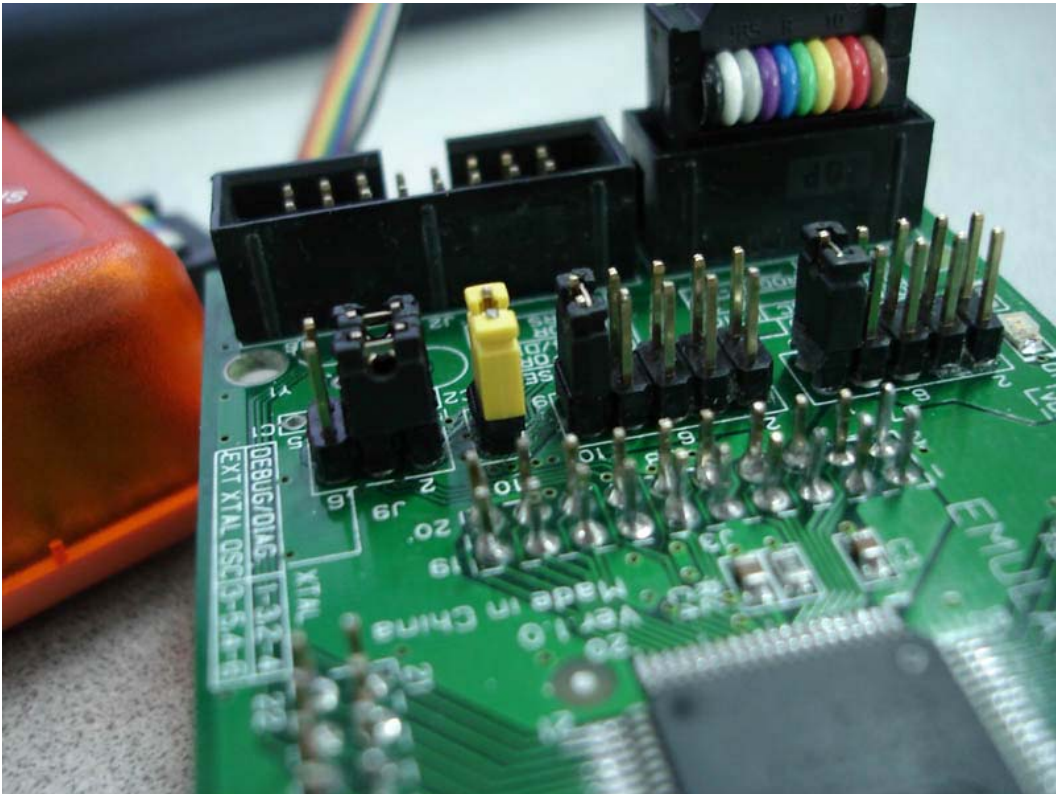


图 6-14　MINICUBE 通过仿真板连接诊断板时的跳线设置

78K0/KF2 的仿真板上有很多引出的针脚，使用跳线帽时要注意，下表列出了跳线的详细连接情况。

表 6-1　MINICUBE 仿真板的各种跳线组合

操作模式	J7/J8	J9	J10	说明
自检方式	9—10	1—3, 2—4	短路	VDD 由 MINICUBE 提供
ICE 方式调试 (内置振荡器)	5—6	1—3, 2—4	开路	VDD 从 Vdd 引脚 或者电源插座提供
ICE 方式调试 (外部振荡器)	5—6	3—5, 4—6	开路	VDD 从 Vdd 引脚 或者电源插座提供
Flash 烧写 (NEC 专用烧写器)	1—2	1—3, 2—4	短路	VDD 由编程烧写器提供
Flash 烧写 (简易烧写器)	5—6	3—5, 4—6	短路	VDD 从 Vdd 引脚 或者电源插座提供

下面是仿真板的正面示意图，四个角上比较大的黑色圆点是安装孔，其余的小一些的黑色圆点都是引出的针脚。其中右下角的Y1，C1，C2 器件并未安装。

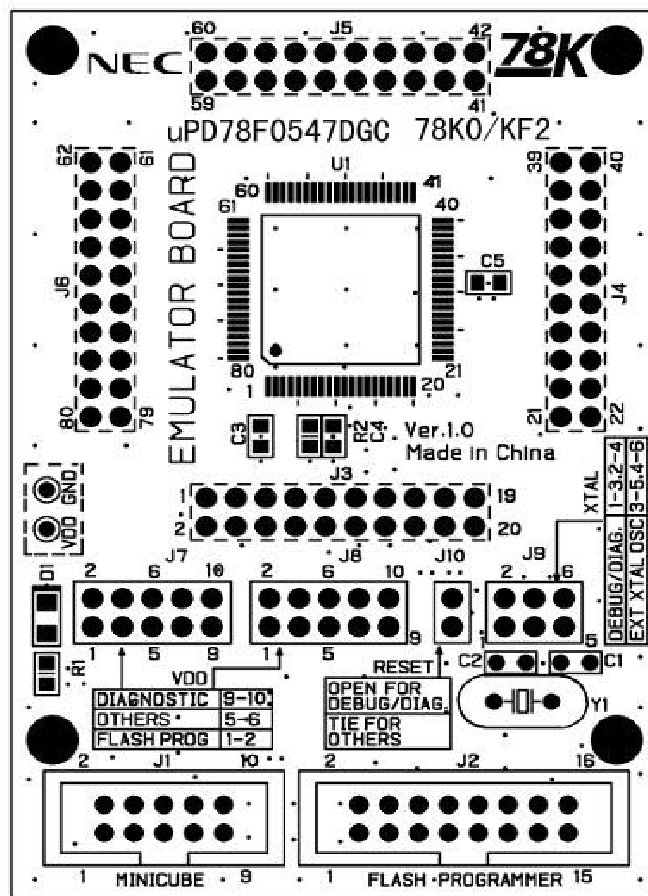


图 6-15　MINICUBE 仿真板的接线图和跳线设置说明

其中右下角 16 针的J2 是PG-FP4 专用的Flash-Programmer接口，如果想要使用简易编程器FPL3 连接到仿真板，那么可以用散线连接到芯片的引脚上，或者加装转接头。参看文件《PG-FP3 Flash Memory Programmer User’s Manual》（U17454E）。

注：FPL3 是 78K0/KX2 以及 78K0/FX2 的编程器。

6.2.2　通过排线连接目标板

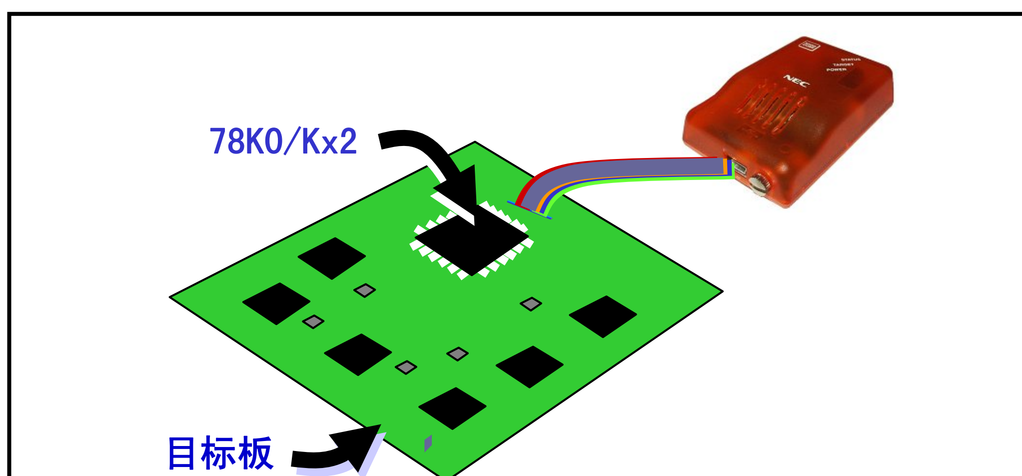


图 6-16　MINICUBE 用排线连接到目标板的示意图

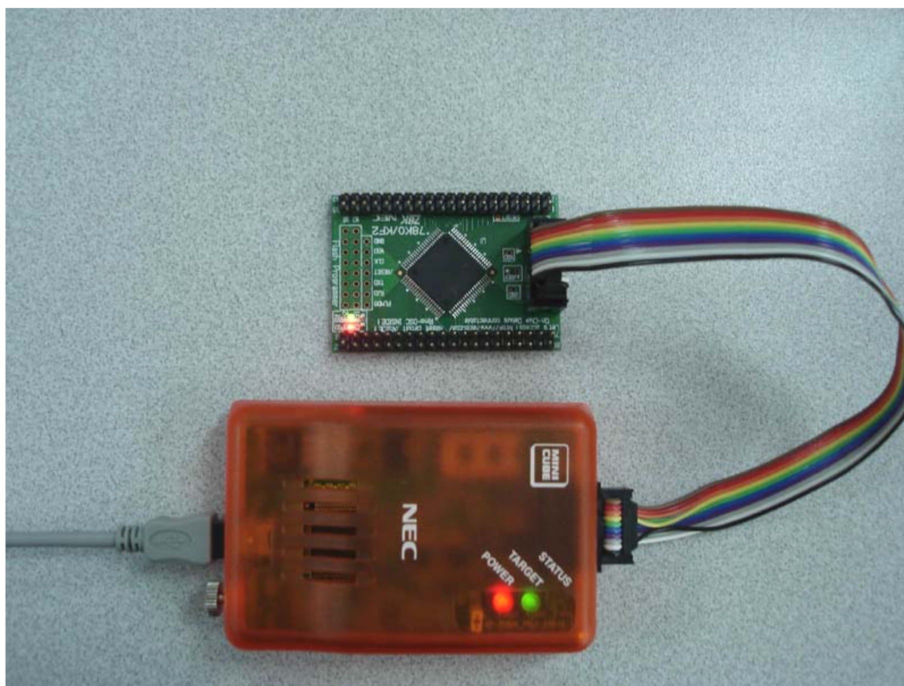


图 6-17　MINICUBE 用排线连接到目标板的实物图

这种连接方法适用于MCU已经焊接在目标板上的情况，需要在目标板上提前留出对应的通讯接口。

6.2.3　通过仿真板以 ICE 方式连接到目标板

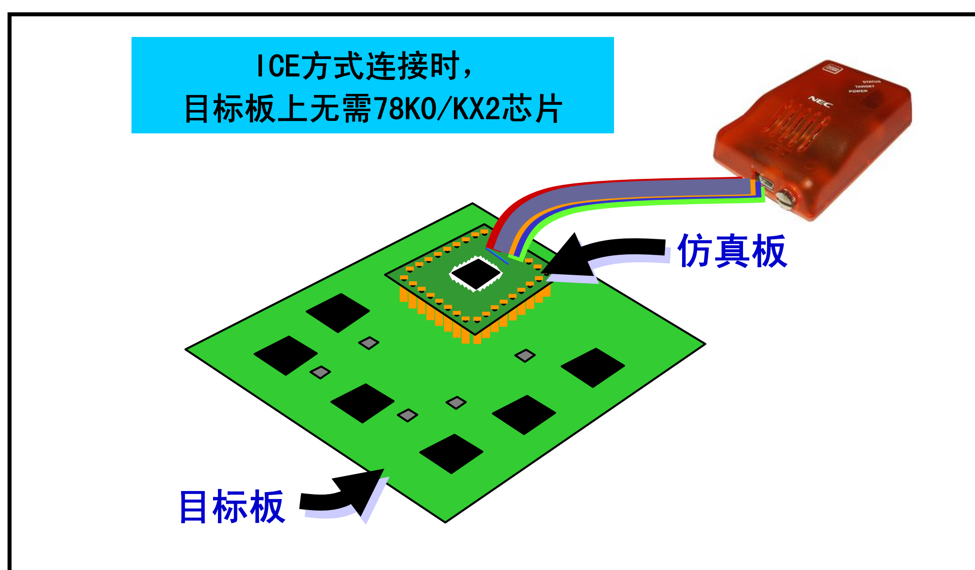


图 6-18　MINICUBE 通过仿真板用 ICE 方式连接到目标板的示意图

这种方式非常的方便，用户可以随意更换MCU。NCT版本的MINICUBE设计了类似ICE连接方式。采用ICE的连接方式后，实验箱上的MCU万一损坏，那么只需要换一个仿真头即可。

6.3　NCT-MINICUBE78K0 功能详述

1.　产品规范

通过USB接口 5V供电，最大消耗电流 500 毫安（此时USB需要有独立电源）。目标接口的电源由目标设备提供的时候，标准消耗电流为 8 毫安（除去目标电路的功率消耗。）

2. 目标接口规范

目标系统的时钟可以用目标板的晶振，或者MINICUBE上的默认 5Mhz时钟。目标系统时钟范围根据目标设备的规定，比如目标型号为 78K0/KX1+，在 2.7V供电时振荡器频率最低要求 2Mhz。目标接口电压允许范围为 2.0-5.5V。当目标系统有Flash存储器时才需要Flash编程接口，从ID78K0-QB写入目标Flash时需要使用接口信号FLMD0。

3. 调试功能：

（1） 片上 ROM/Flash 安全（On-chip ROM/Flash security）：10 字节的 ID 校验
（2） 事件触发的断点功能（Event-triggered break）：执行前断点数量 1 个（软件断点时为 0 个），访问断点数量 1 个。
（3） 软件断点（Software break）：2000 个
（4） 强制断点（Forced break）：有
（5） 自动步进（Slow motion）：有
（6） 伪实时 RAM 监测（Pseudo real-time RAM monitor）：RAM 区域：16 字节（可以有 8 点 1 位单元，通过临时断点实现）
（7） 动态存储器修改（DMM）：有（通过临时断点实现）
（8） 寄存器操纵功能（Register manipulation）：有
（9） 屏蔽引脚（Mask）：RESET
（10） SP 可以超出保护区（SP setting overlook protection）：是
（11） 测量程序运行时间（Execution time measurement）：不支持
（12） 程序跟踪功能（Trace）：不支持

4. 硬件的安装

（1） 使用兼容 USB2.0 的 MINI-B 接口。
（2） 目标设备的时钟插座。

给MINICUBE内部的时钟插座上安装振荡器时MINICUBE要断电。当调试器的Configuration界面“Main Clock”参数选择“System”时，时钟插座上最好不要安装振荡器。安装的办法请参阅《NCT-MINICUBE78K0 On-Chip Debug Emulator User’s Manual》（U17029E）。

5. 关于自检的 OCD Checker

用户在给诊断板下载程序时，如果违反了MINICUBE的限制，使用了OCD占用的单元，那么OCD Checker和ID78K0-QB都将无法运行。这时候需要对诊断板进行恢复：连接FP4编程器，擦除片内的程序。

6.4 NCT-MINICUBE78K0 的使用限制

1. RESET-IN 信号的持续时间小于 50us 时，目标设备的 RESET-OUT 的下降沿要比 RESET-IN 的下降沿延迟 15us。
2. RESET-IN 信号的持续时间大于 50us 时，目标设备的 RESET-OUT 的上升沿要比 RESET-IN

的上升沿延迟 25us，RESET-OUT 的下降沿要比 RESET-IN 的下降沿推迟 15us。

3. 使用两个端口（不是 X1X2）进行 OCD 调试的时候，这两个端口不能被用户用于其它用途。
4. 使用 MINICUBE 对目标芯片进行调试的时候，目标芯片必须使用 MINICUBE 提供的 RESET 信号。
5. Flash 通过自编程烧写用户程序时，这时要保证振荡器符合自编程操作的要求。
6. 仿真过程中 FLMD0 的输出保持为高电平，请确保给此引脚连接下拉电阻，使用此端口时，应该按照高/高阻抗处理，而不是高/低阻抗。
7. 使用诊断板进行程序检测时的限制

   (1) 当 MINICUBE 使用 Ring-OSC 时，MCU 中的 BUG 可能会导致通讯异常。调试器为了防止这个问题，在 Break 断点发生时强迫切换到主时钟运行，当程序执行时，又切换到 Ring-OSC 时钟。所以，在断点时 MINICUBE 总是以主时钟运行。

   (2) 因为诊断板上的 MCU 的 BUG 引起内部 RESET 时，会产生 Fail-safe 中断（例如堆栈没有进行初始化就开始使用堆栈而引起的中断）。

[解决办法]：为了防止看门狗定时器引起的内部RESET，停止看门狗或者保证看门狗的运行。

注：关于使用 MINICUBE 调试时占用用户资源的详细情况见于附录 B。

第 7 章　集成调试环境

本章介绍如何使用ID78K0-QB for MINICUBE调试用户程序。

7.1　启动 ID78K0-QB for MINICUBE

点击“开始→程序→NEC Tools32→ID78K0-QB for MINICUBE” 图标，将启动ID78K0-QB for MINICUBE软件。

进入开发调试环境之后，自动弹出配置对话框，由用户根据自己的目标板配置来设置ID78K0-QB for MINICUBE执行时的环境。配置界面如下：

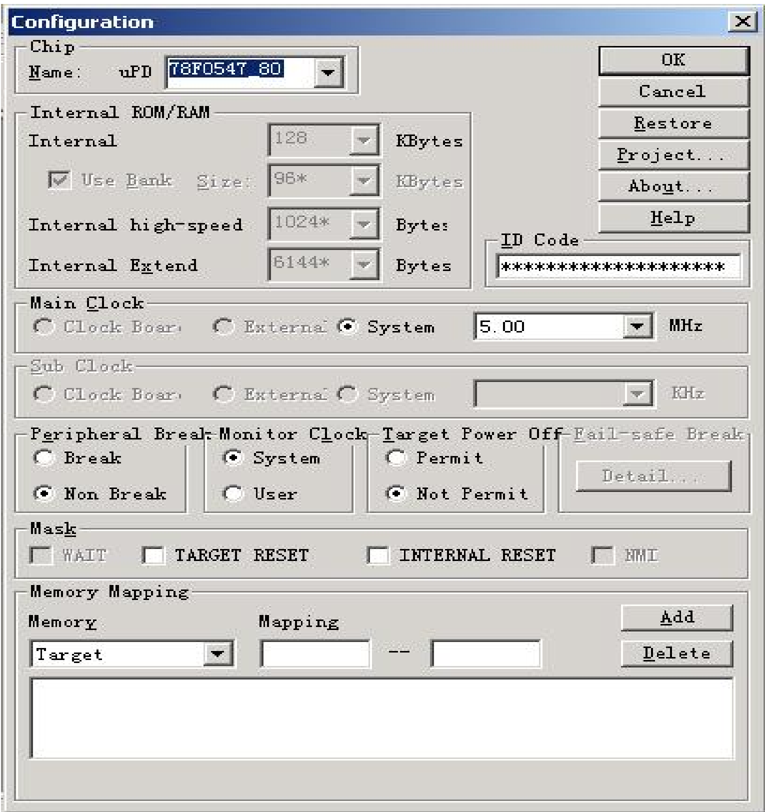


图 7-1　ID78K0-QB 的 Configuration 对话框

Chip　　ID78K0-QB for MINICUBE 使用的芯片型号，下拉框提供了已经安装的所有芯片。

Internal ROM/RAM　　执行时芯片的内部 ROM 和 RAM 大小。默认值根据所选的芯片而变化。修改内存大小的选项是用于软件仿真的。现在使用的是硬件仿真器，所以值都是灰色的，不可更改。

Main Clock　　CPU 的主时钟和子时钟的频率值。

Memory Mapping　　外部内存设置，下拉框提供了可添加的内存类型。

这里如果打开ID78K0-QB for MINICUBE的同时，PM+软件是打开的，那么这个Configuration对话框中的Chip是灰色的，无法更改，自动认为是PM+的当前工程中使用的芯片型号。

设置完成后，点击“OK”按钮，将显示ID78K0-QB for MINICUBE的主窗体。

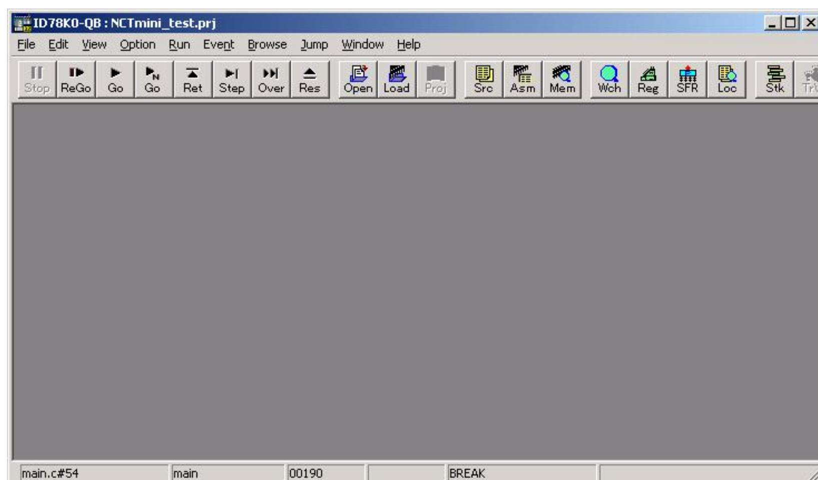


图 7-2　ID78K0-QB 软件主界面

此界面中载入工程的“Proj” 按钮是灰色的，因为此时已经有工程载入了，在标题栏可以看到有“NCTmini_test.prj”的字样。没有工程载入时，“Proj”按钮是可选的，而“Src”按钮是灰色的。

7.2　界面介绍

首先是ID78K0-QB for MINICUBE的主界面各个窗口介绍。

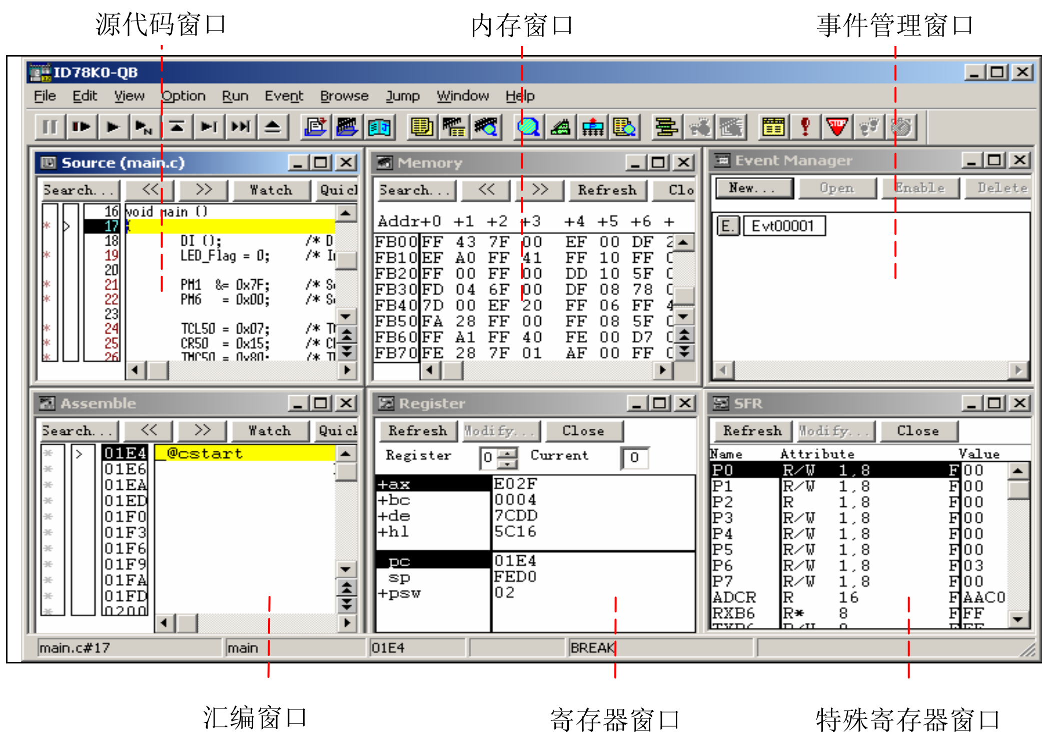


图 7-3　ID78K0-QB 的各个窗口

其中调试工具栏的按钮说明如下，基本上我们绝大多数的调试工作都可以用这些按钮来实现。灰色的是MINICUBE所不支持的功能，如果需要使用这些高级功能选项，请使用IECUBE等硬件。



图 7-4　调试工具栏的按钮说明

7.2.1　下载目标文件

目标文件就是由用户程序编译生成的二进制文件，扩展名是*.lmf。关于编译目标文件的方法，详见第 5 章。

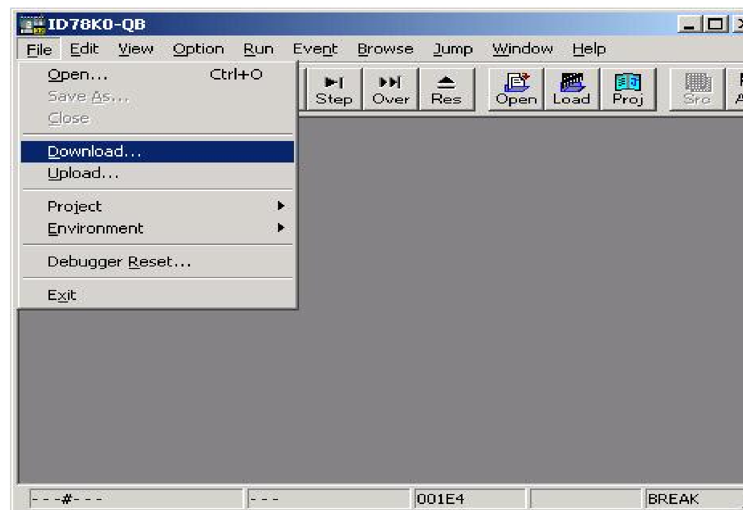


图 7-5　加载目标文件

要加载目标文件，选择“File”菜单中的“Download…”，打开Download对话框。

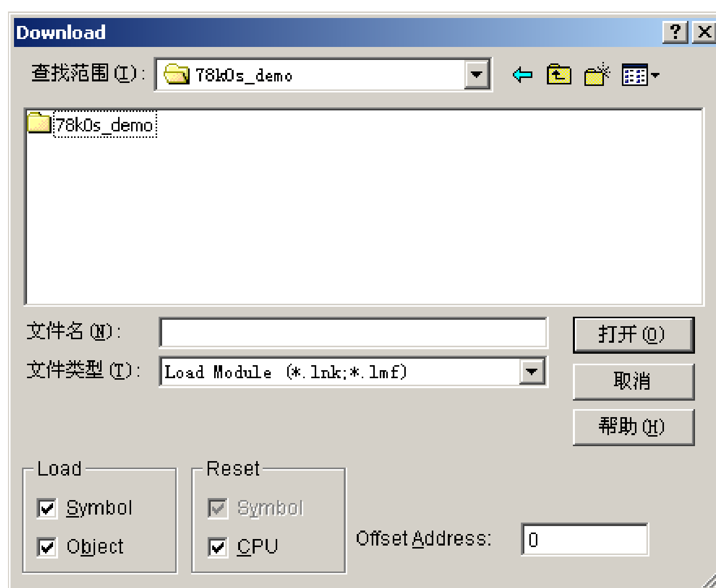


图 7-6　Download 对话框

选择要下载的目标文件（*.lmf），点"打开"按钮，该文件就被加载到ID78K0-QB for MINICUBE中。

此时相当于编程烧写器的功能，下载完成后，断开和MINICUBE的连接后，目标板可以独立运行。

7.2.2　加载源文件

源文件是工程中包含的C和汇编文件。

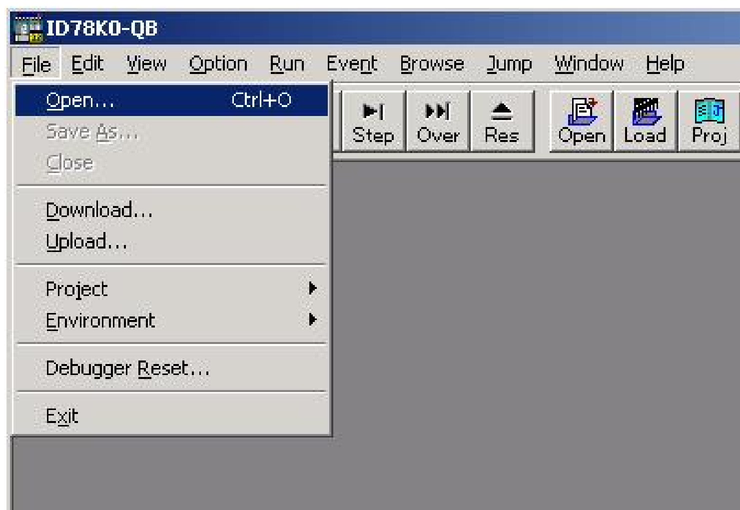


图 7-7　加载源文件

要加载源文件，选择"File→Open..."，打开Open对话框。

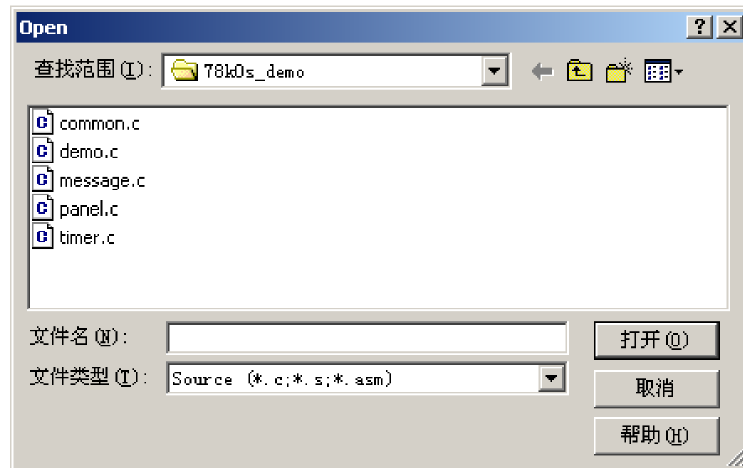


图 7-8　Open 对话框

选择要加载的源文件，点“打开”按钮，该文件就被加载到ID78K0-QB for MINICUBE中。如下图：

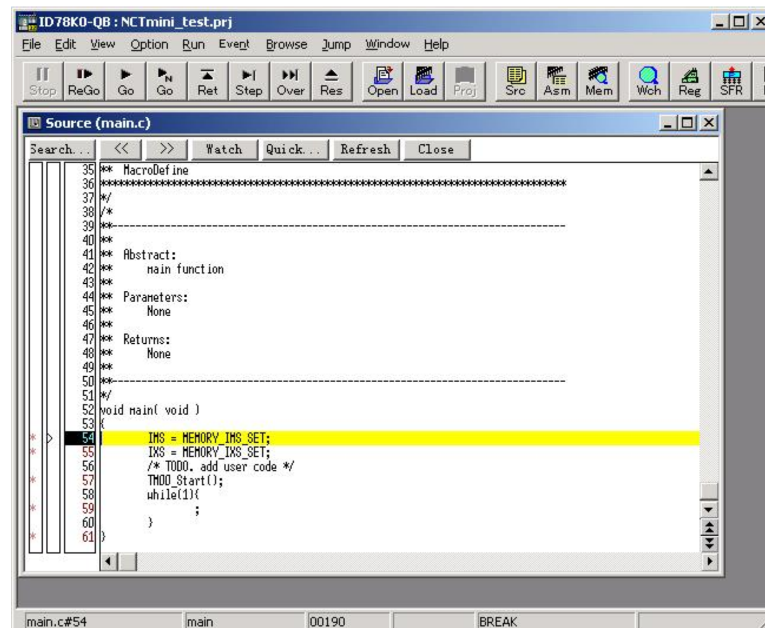


图 7-9　ID78K0-QB for MINICUBE 源程序窗口

在ID78K0-QB for MINICUBE的源代码窗口中，右栏显示代码，左栏显示调试信息。

调试信息栏的数字和符号意义如下：

*　　　表示本行是指令行。可以在有“*”的行设置断点。

\>　　　表示当前程序执行的指令行。当前行显示为黄色。

数字　　表示行号。

在源文件窗口的时候，如果原来的程序使用C语言编写，又想观察对应的汇编语句，那么可以在源文件窗口任意地方点击右键，再弹出的菜单上选择“Mix”，就可以看到了。

7.2.3　加载工程文件

工程文件（*.prj）包含了调试一个工程的所有信息，包括断点、事件、加载文件、窗口位置和各种设置。通过加载一个预先保存好的工程文件，可以方便的恢复上次调试的所有环境。

要加载工程文件，选择“File”菜单中的“Project → Open”，打开Open对话框。

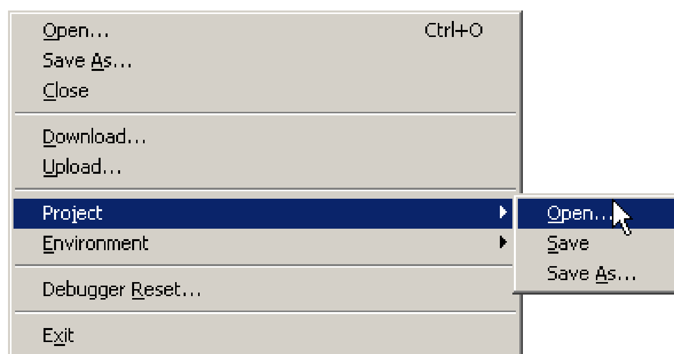


图 7-10　加载工程文件



图 7-11　Open 对话框

选择要加载的文件名，点击“打开”按钮即可。

7.2.4　在源文件中设置断点

每行程序左边都有一个“*”标识，鼠标左键点击“*”号，就在该行设置了断点。这样，行号左边的“*”标识变为蓝色的“B”，表示成功设置了一个软件断点。如果该行不是当前运行命令行，那么显示为红色。如果断点设置在当前运行命令行，那么此行仍然显示为黄色。

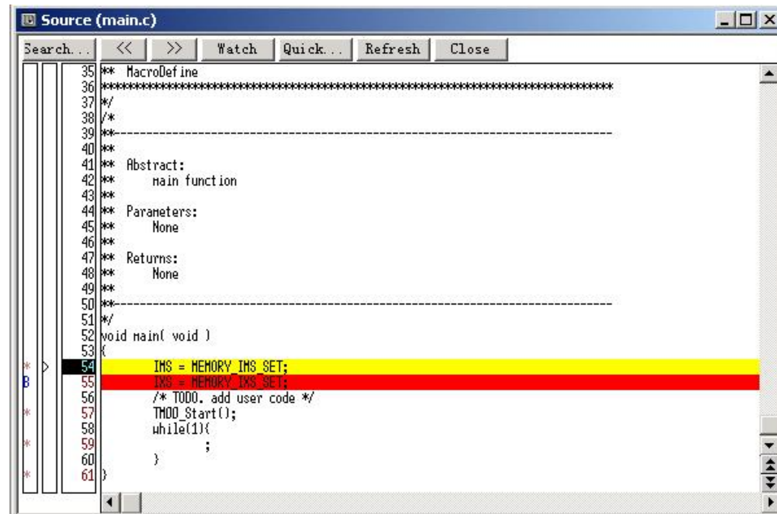


图 7-12　设置断点

7.2.5　执行程序

要执行程序，点击主窗体工具栏上的执行按钮，或者选择“Run→Go”。

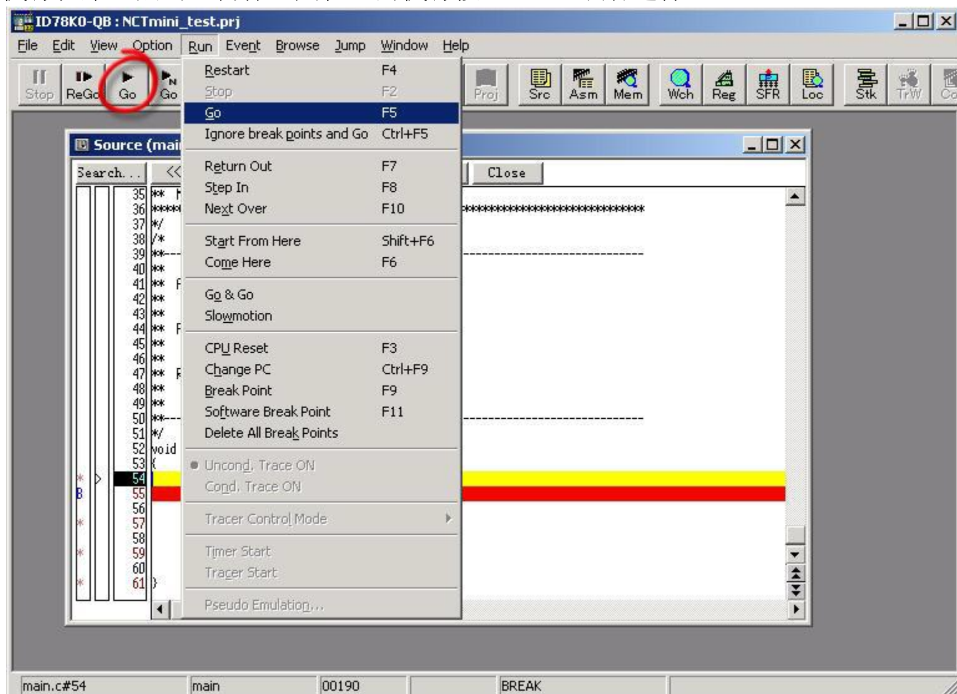


图 7-13　执行程序

程序执行到断点处将会停止。要去掉断点，只需左键点击“B”标号即可。

还有一个无视断点运行的功能，点击“Run→Ignore break points and Go”，那么不论有多少断点，程序都会连续运行，完全忽略断点的作用。

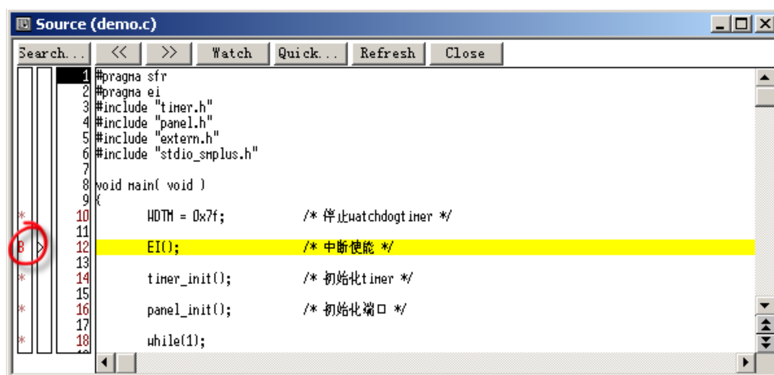


图 7-14　清除断点

7.2.6　单步执行

点击主窗体工具栏上的Step In按钮，或是选择“Run→Step In”，进行单步执行。

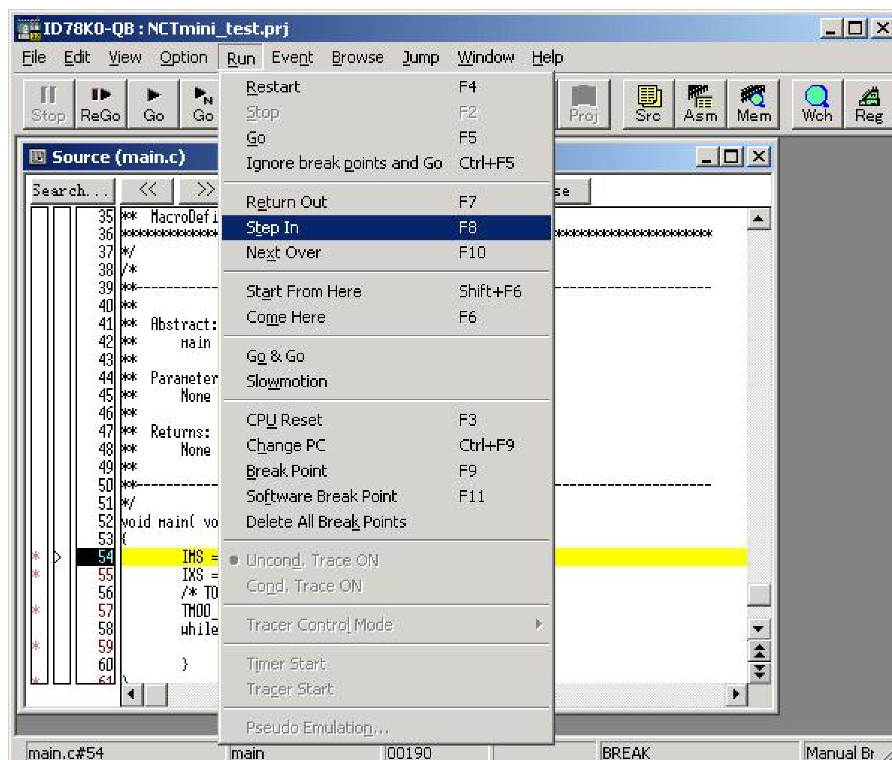


图 7-15　单步执行

Step In按钮把源文件的一行作为一步来执行，遇到函数调用，则进入函数的代码，逐行执行。

Next Over按钮也是把源文件的一行作为一步来执行，但在函数调用时，整个函数作为一条语句执行。

7.2.7　停止执行

在程序执行期间，按停止按钮，或是选择“Run→Stop”，可以停止程序的执行。

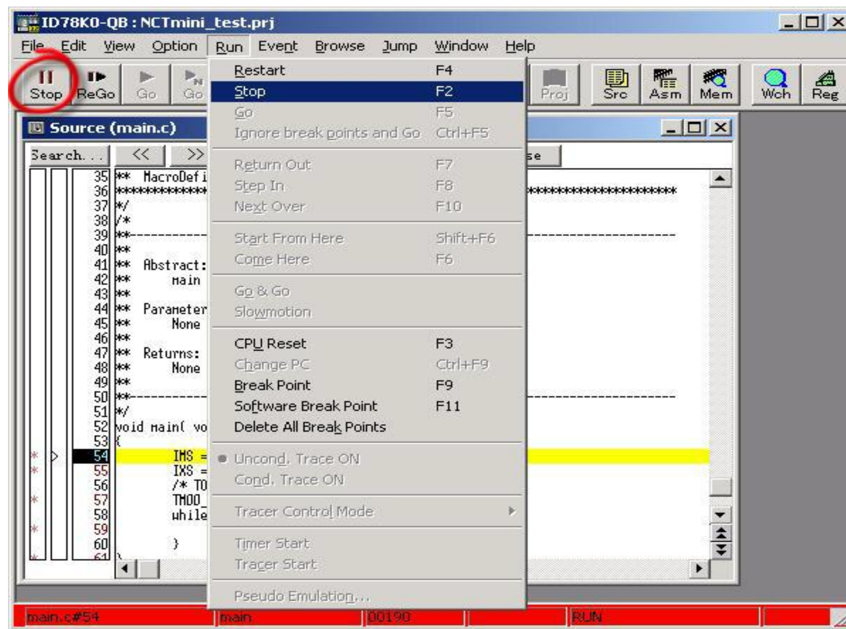


图 7-16　停止执行

7.2.8　复位

无论程序是否在执行，都可以进行重启操作。重启的方法是按下复位按钮，或是选择“Run→CPU Reset”。

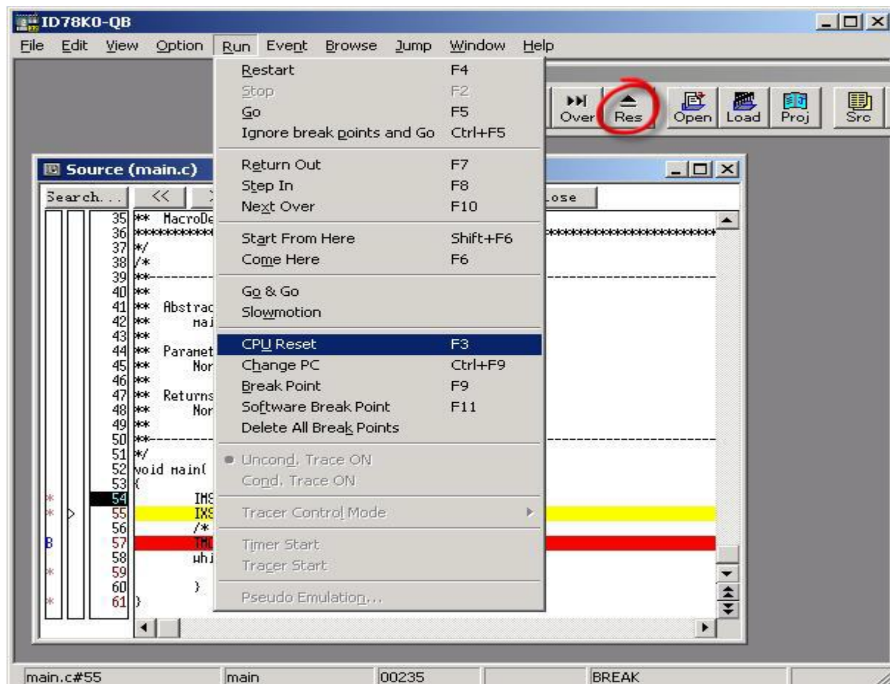


图 7-17　重启

复位操作将使CPU和各外围Macro都恢复初始状态。所有寄存器恢复初始值。

建议不要在程序运行时直接进行复位操作，而应该先停止程序，后复位。

7.2.9　观察和修改变量值

开启 Watch 窗口：

点击主窗体工具栏上的Watch按钮，或是选择“Browse→ Watch”，开启Watch对话框。

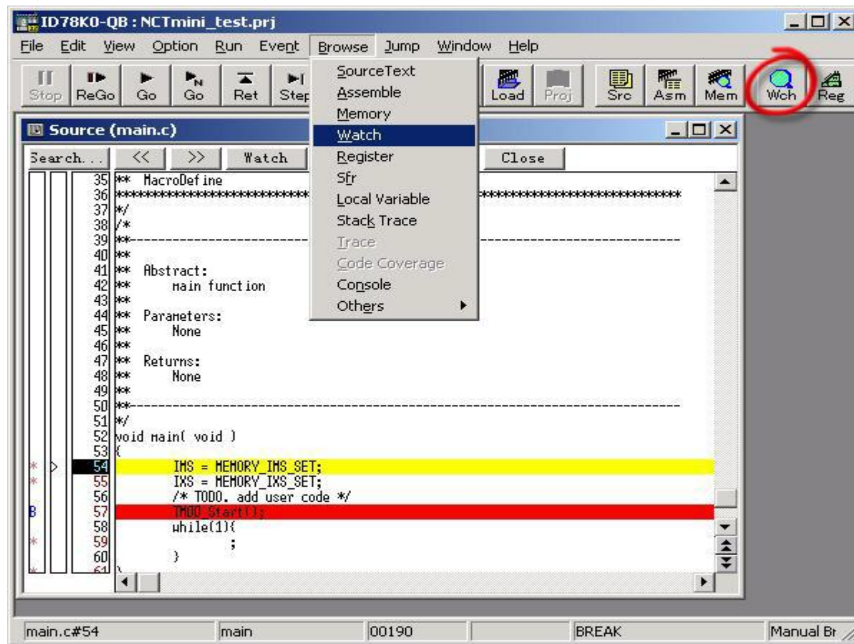


图 7-18　打开 Watch 窗口

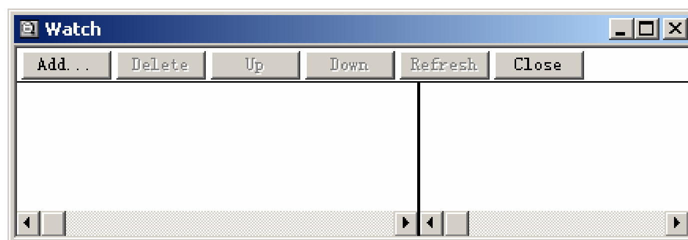


图 7-19　Watch 窗口

第一次打开的Watch窗口内容为空。

观察变量：

在图 5-17 中点击Watch窗口的“Add…”按钮，弹出添加变量对话框。

其实还有更方便的办法可以进入Watch窗口，就是在想要观察的变量上点击鼠标右键，在弹出的菜单上选择“Add Watch”就可以看到下图。

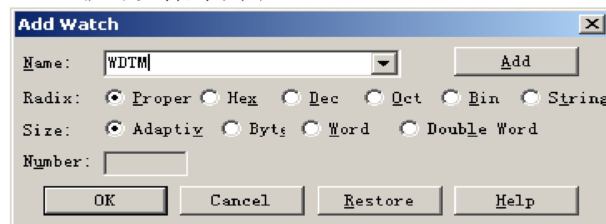


图 7-20　添加变量

在对话框的Name栏填入变量名，选择“OK”，就可以观测到该变量的当前值。随着程序的执行，变量的值也会随时刷新。

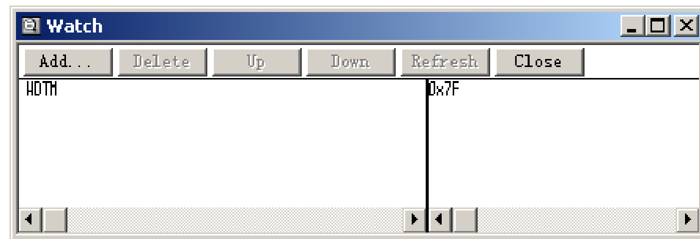


图 7-21　观察变量

修改变量值：

在Watch窗口的右侧区域，输入新的值即可修改该变量的值，按回车键确认修改。通过这种操作，可以在调试中很方便的修改变量的数值，而不必重新对源程序进行修改和编译。

数值被修改时显示为红色，按下回车键确认后，数值显示为黑色。

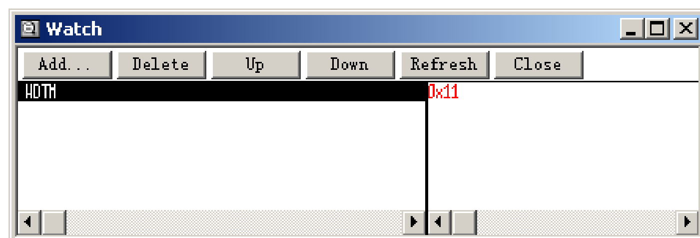


图 7-22　修改变量值

7.2.10　观察和修改寄存器值

开启寄存器窗口：

点击寄存器窗口图标，或是选择“Browse→ Register”，开启寄存器窗口。

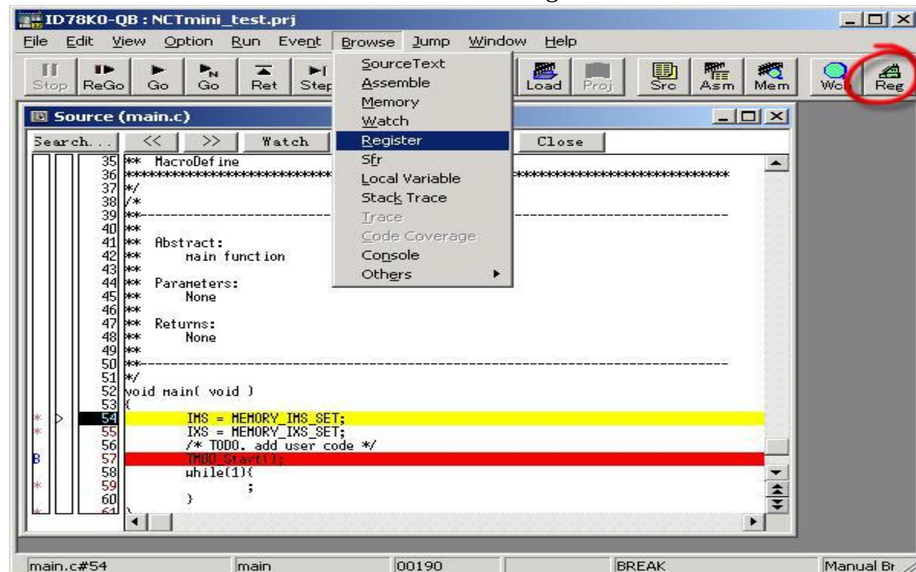


图 7-23　打开寄存器窗口

观察寄存器值：

双击带“+”的寄存器的名字，可以看到寄存器中各个位的值。

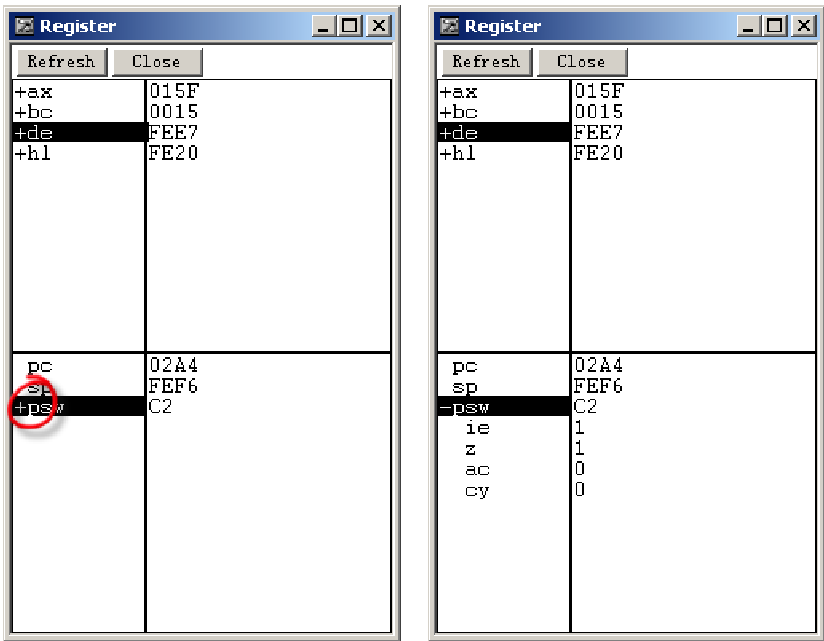


图 7-24　寄存器窗口

修改寄存器值：

在寄存器窗口，可以修改寄存器的值，修改后的值显示为红色，按回车键确认后，数值显示为黑色。

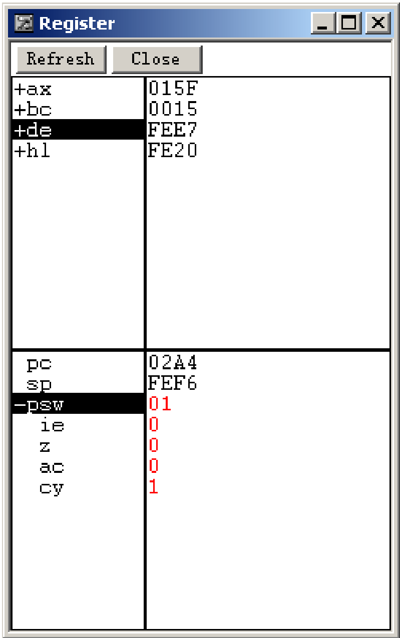


图 7-25　修改寄存器值

7.2.11　观察汇编代码

ID78K0-QB for MINICUBE可以显示程序的汇编代码。点击汇编窗口按钮，或是选择“Browse→Assemble”，打开汇编窗口。

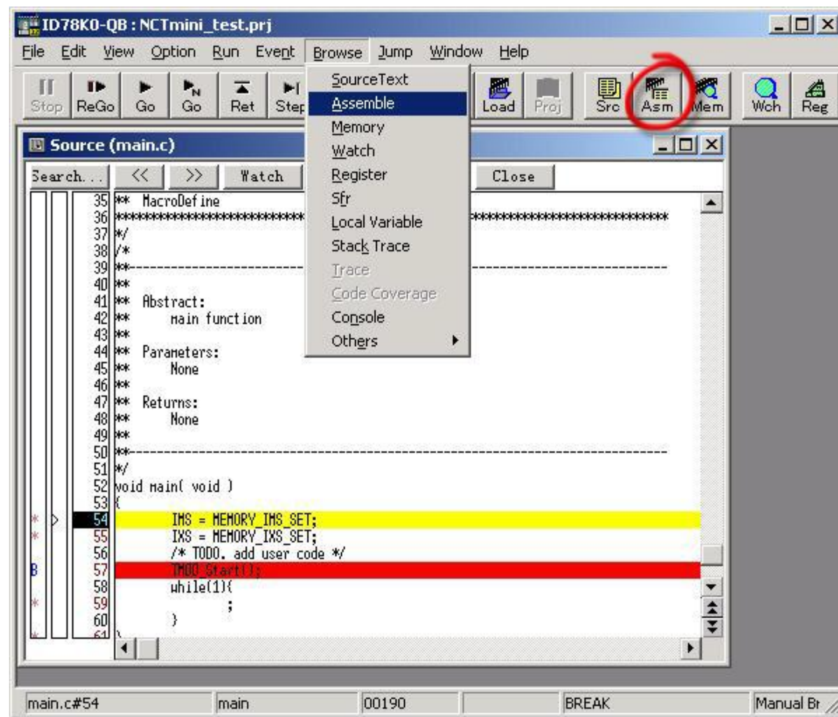


图 7-26　打开汇编窗口

或者在Source窗口点击鼠标右键，然后在弹出菜单上选择“Assemble”也可以打开汇编窗口。

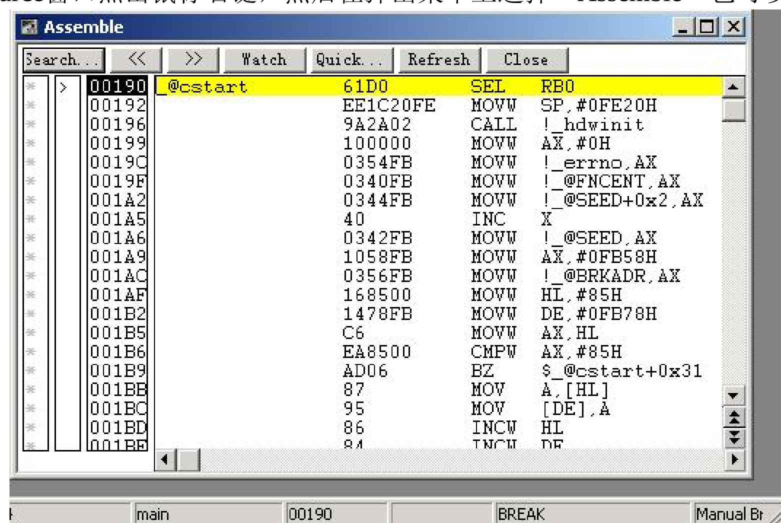


图 7-27　汇编窗口

与源代码窗口类似，左侧是调试信息栏，其中“*”符号表示汇编指令行，可以在有“*”的行设置断点；“>”符号表示当前执行的指令行，显示为黄色；“数字”表示指令地址。

7.2.12　设置汇编断点

在地址行左边的“*”区域点击鼠标左键，可以加入一个断点，断点处有一个蓝色的“B”标识。和 7.2.4 节中的Source窗口一样，被设为断点的行如果不是当前命令行，那么显示为红色。

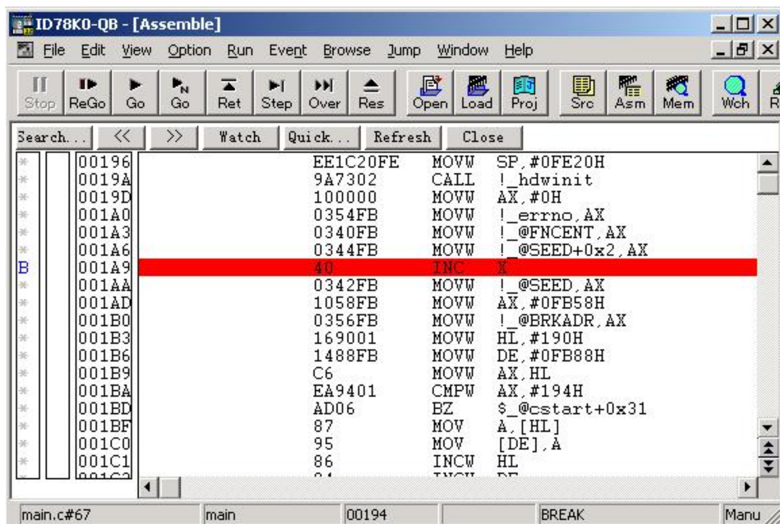


图 7-28　设置汇编断点

点击主窗体工具栏上的按钮执行，程序执行到断点时停止。标识了“B”的黄色指令行表明程序执行的当前位置。

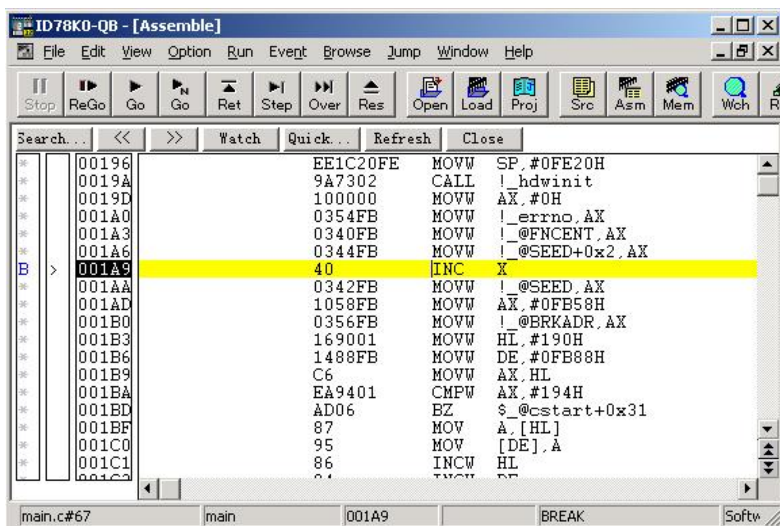


图 7-29　执行到断点

7.2.13　修改汇编代码

与Watch窗口修改变量值类似，在Assemble窗口可以修改指令，修改的指令为红色，按回车键确认修改。

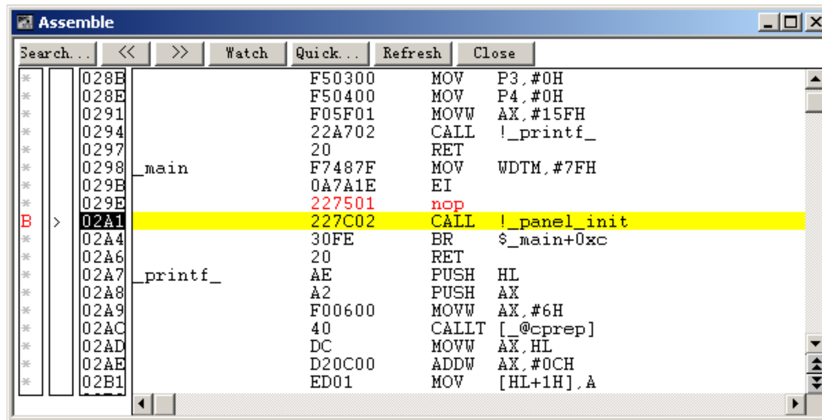


图 7-30　修改汇编指令

7.2.14　退出 ID78K0-QB for MINICUBE

要退出ID78K0-QB for MINICUBE，选择File菜单中的Exit，出现退出提示对话框。



图 7-31　退出对话框

询问是否将ID78K0-QB for MINICUBE当前的调试环境设置保存为一个调试工程文件。如果点击“是（Y）”，下次启动ID78K0-QB for MINICUBE时，可以使用已经保存的环境设置。

7.3　高级调试功能

本部分内容是在 7.2 的基础上，介绍一些ID78K0-QB for MINICUBE提供的高级调试功能，可以帮助用户更方便的调试程序。

7.3.1　Event（事件）

在ID78K0-QB for MINICUBE中，Event被定义为各种调试动作的触发条件。

基于Event可以设置断点，Event还可以作为启动或停止Break的条件等。通过Event管理器（Event Manager）对Event进行管理。

选择“Event→ EventManager”，打开Event Manager窗口。

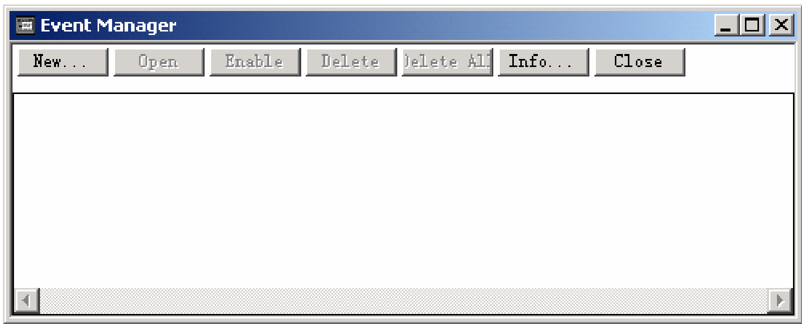


图 7-32　Event Manager 窗口

在Event Manager窗口中，点击“New…”按钮，出现New Event窗口，选择“Event…”，打开Event对话框。



图 7-33　New Event 界面

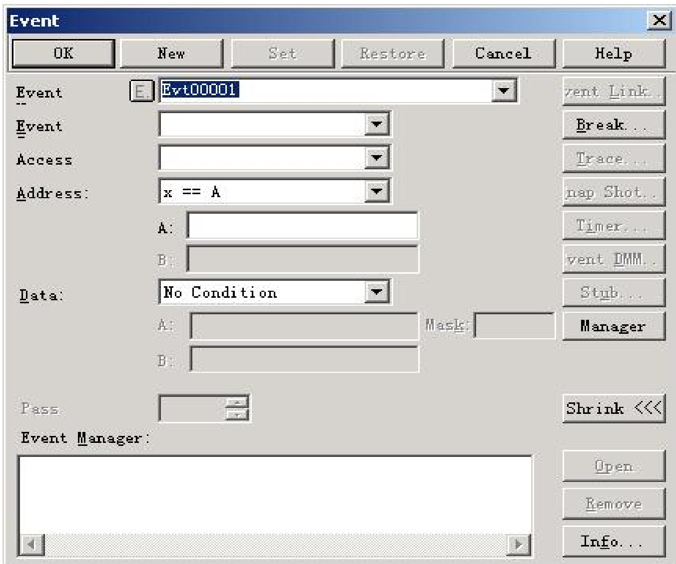


图 7-34　Event 窗口

Event Name	新的 Event 名称。
Event	Event 的触发条件。
Access Size	传输数据的位宽条件。位宽不符合，将不能触发 Event。
Address，Mask	地址范围条件，地址的 Mask 值。
Data，Mask	传输的数据条件，数据的 Mask 值。

Pass Count　　　　条件重复次数。表示发生多少次触发条件，事件才会被触发。默认为1，即第一次满足条件，就触发事件。

点击“Set”按钮，Evt00001 就设置好了，可以在Event Manager窗口看到Evt00001。如果在Source窗口或Assemble窗口中设置了事件，可以在行前看到“E”标识。

同一类型的Event最多可以设置 256 条。

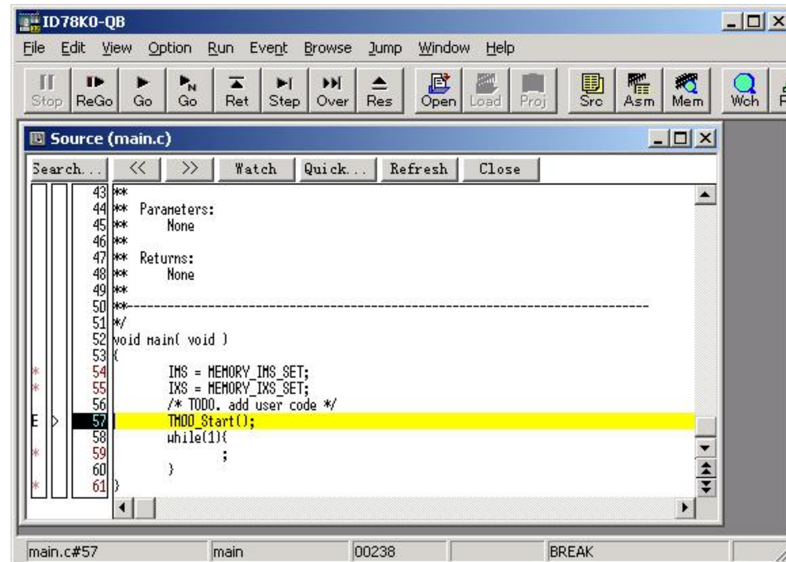


图 7-35　设置有 Event 的 Source 窗口

7.3.2　Break（断点）

除了 7.2.12 节中介绍的设置断点的方法。还可以用Event来设置断点，这样断点的触发条件可以定义的非常灵活。

先用 7.3.1 节的方法设置一个Event。然后在图 7-33 的New Event界面中，点击“Break…”按钮，打开Break窗口。

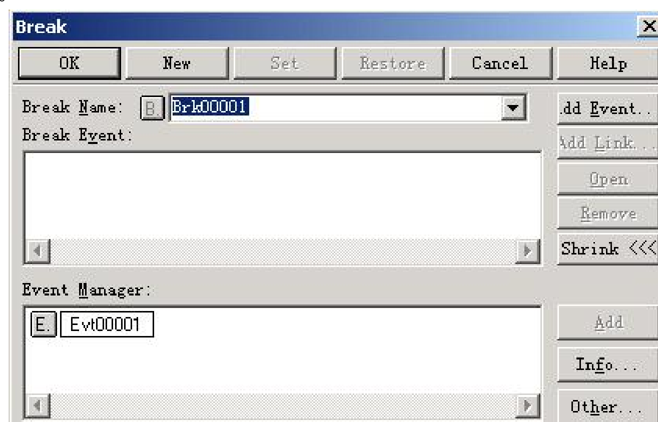


图 7-36　Break 窗口

通过拖动的方法，选择Event作为断点条件。

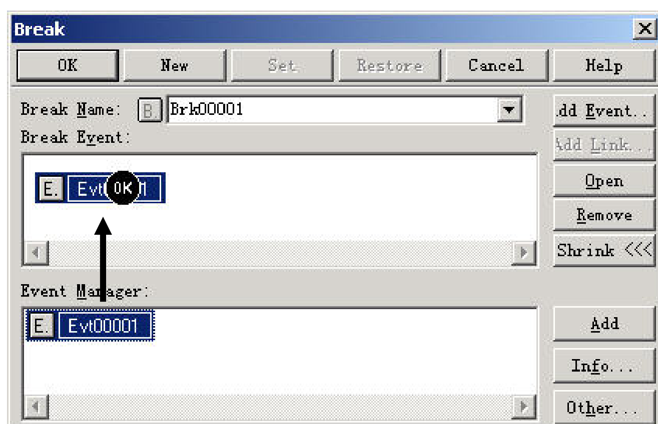


图 7-37　将事件设置为断点条件

点击“Set”按钮，断点Event条件设置完成，Break窗口Event Manager列表框中，Brk00001 图标变为红色。

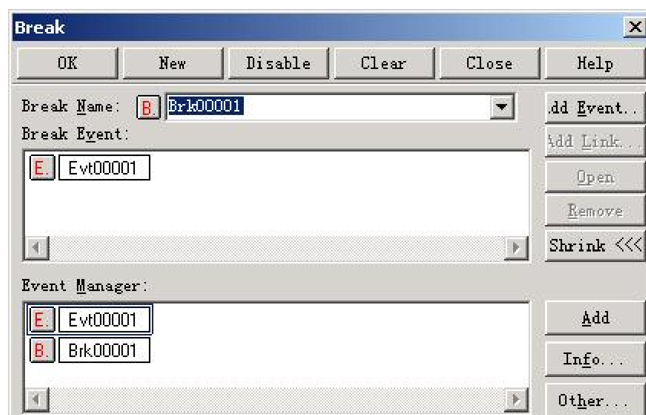


图 7-38　断点设置完成

7.3.3　两个比较重要的选项

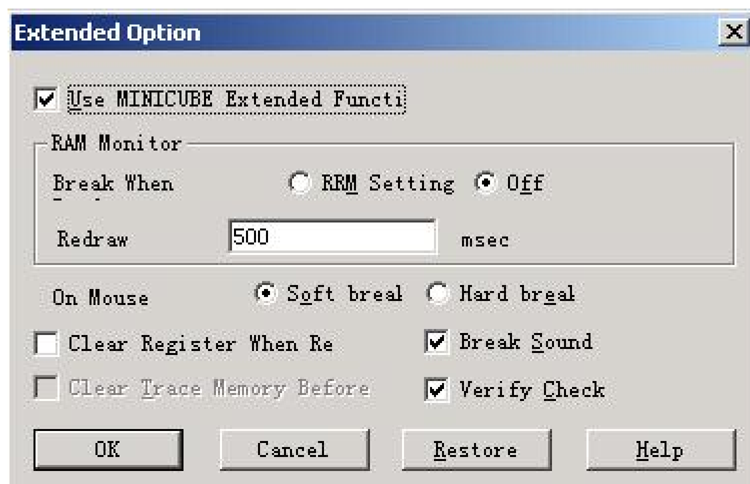


图 7-39　扩展选项的对话框界面

这个扩展选项主要是用来设置MINICUBE的一些扩展功能，默认是关闭的，需要激活才能使用。

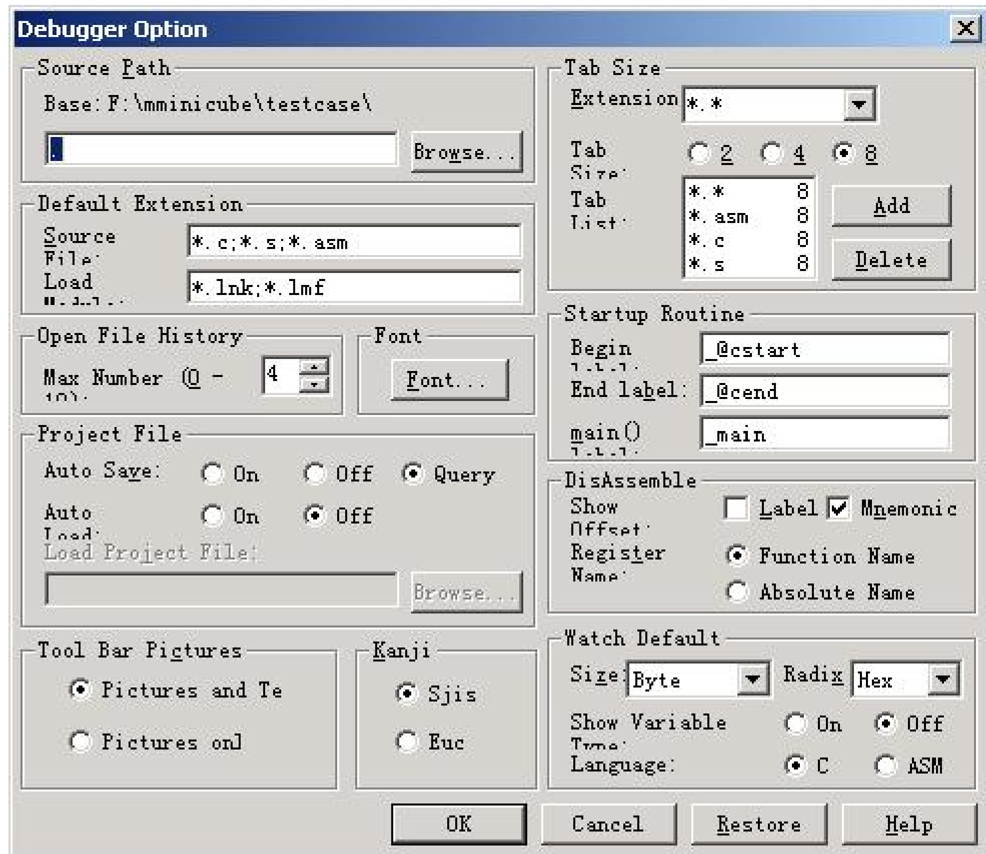


图 7-40　调试选项的对话框界面

有很多常用的参数设置都在这里完成。如果不熟悉调试工具栏的按钮，那么在“Tool Bar Picture” 选择“Picture and Text”，这样的话每个按钮下方就有对应的文字提示。

附 录 A　OCD Checker 的使用

注：进入这个程序时要注意仿真板的跳线必须放在自检模式。

一、 OCD Checker 的操作界面

检查的项目包括目标晶振，目标板电压，目标RESET，MINICUBE版本，目标设备型号，对Flash 存储器的下载、运行、擦除测试。

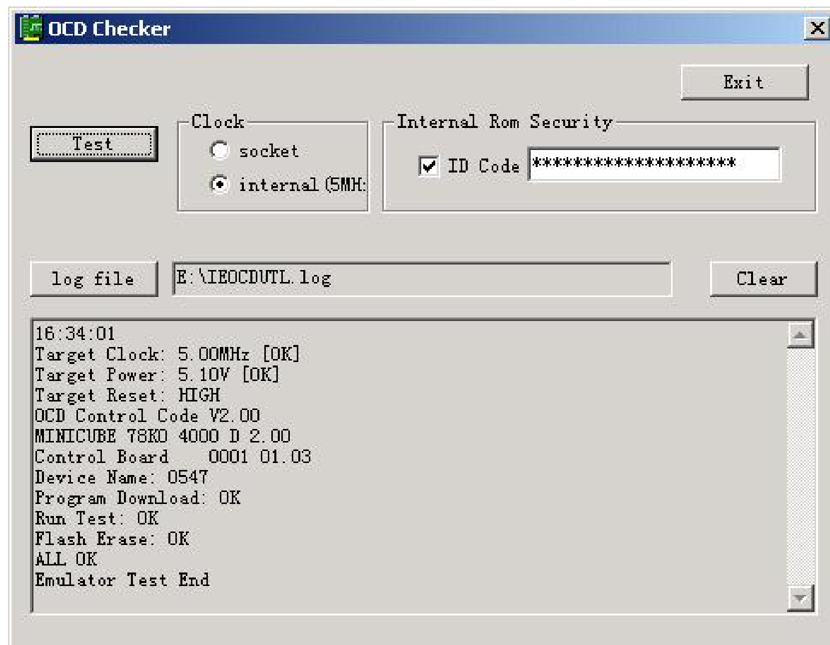


图 1 Security ID 错误时的提示界面

二、 一些错误的解决办法：

1. “通讯错误”

   MINICUBE 被禁用或者连接不正常。

2. “驱动错误”

   (1) MINICUBE 没有反应：

   检查 USB 连线，断开 USB（断电）然后再接上，然后重启计算机。

   (2) 调试器正在运行

   OCD Checker 和 Debugger 不能同时运行。

   (3) USB 驱动不正常

   (4) 调试器软件安装不正确。

3. “目标设备的电源关闭”

   诊断板检测不到电压。如果诊断板的电源指示灯亮，检查连接线。

4. “目标时钟”：0.00MHz

   MINICUBE 插座上的时钟无动作，此时要改用内置时钟。插座上有外接振荡器的时候不要选择内置的 5MHz 晶振。

5. “ID 错误”

   默认 ID Code 为 10 个 FF；如果 ID 错误，则显示如下信息：

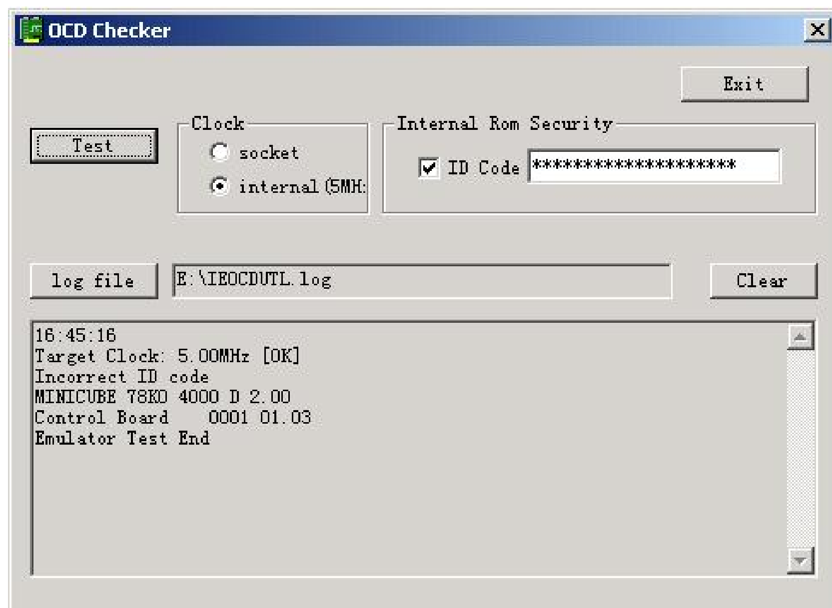


图 2 Security ID 错误时的提示界面

6. “ROM 命令错误”或者“监控命令错误”

   MINICUBE 和目标板的连接异常，检查步骤同 B。

7. “读写 Memory，数据校验错误”

   内部 RAM 读取错误，目标设备损坏。

OCD Checker 运行时，会擦除原来的目标程序。可以忽略 FP4 的安全 Flag，但是原来写入的时候选择的参数是“不允许启动区重新编程”，OCD Checker 将无法使用。

更多错误以及解决办法请参考文件《78K0，78K0S Series OCD Checker V1.02 User’s Manual》（ZUD-CD-05）。

附 录 B　使用 NCT-MINICUBE78K0 的注意事项

1. 为 NCT-MINICUBE78K0 保留的区域

（1）Flash 存储器地址

- 0x02,0x03,
- 0x7E,0x7F（使用软件断点时）
- 0x84
- 0x85—0x8E
- 0x8F—0x18F：程序空间标准值
  使用伪实时 RAM 监控功能，则还需要+256 字节
  当设备有 10 个或更多可以通过 16 位的单元访问的 SFR 时，+ n 字节 （n 为超出的寄存器数量*6 字节。）

（2）内部扩展 RAM 区

- 0xF7F0—0xF7FF
  使用伪实时 RAM 监控功能时

（3）内部高速 RAM 区

- 7 个字节作为堆栈区：堆栈的标准值
  使用软件断点时+2 字节，使用伪实时 RAM 监控功能时+7 字节。

2. 0x02,0x03 和 0x8F 以及更高地址的值

0xFF 之外的数值都无法下载到 0x02，0x03 和从 0x8F 开始的程序区内。为了防止对这些值进行误操作修改，在存储器显示中这些地址的值一直都些显示为 0xFF，上载的值同样也是 0xFF。当这些区域被用户程序占用后，OCD 功能将无法进行。

3. 连接参数-go 指定的值

在使用连接参数[-go]时，下列地址保留给 MINICUBE

- 0x02，0x03
- 从 0x8F 开始的程序段空间+1

注：更详细的信息请参阅手册《NCT-MINICUBE78K0 On-Chip Debug Emulator User’s Manual》（U17029E）。

4. 地址 0x7E,0x7F 处的值

这两个地址用来设置软件断点，参看下面的例子，确保这两个地址不会被用户程序占用。

```
例子：SSS    CSEG    AT      07EH;              “SSS”是名称
             DB      0FFH，0FFH
```

5. 地址 0x84 的值

下列设置可以被用于安全 ID 码 通过设置 0x84 地址的值：

- 0x00：MINICUBE 的连接被禁止
- 0x02：如果安全 ID 校验失败，Flash 存储器不能被清除
- 0x03：如果安全 ID 校验失败，Flash 存储器全部被清除

除了 0x02，0x03 之外的值不能被写入地址 0x84，想要设置 0x00 时，需要使用编程器。0x00 被写入后，RESET 后 MINICUBE 和目标板的通讯就被禁止，调试将无法进行。

6. 地址 0x85 到 0x8E 的值

0x85 到 0x8E 的安全 ID 可以在存储器窗口被修改，如果被用户程序覆盖，那么通讯就被停止。调试器重启时，在配置对话框重新设置安全 ID。如果 ID 不匹配，调试软件无法进入。地址 0x85 到 0x8E 是系统预留的安全 ID 区域，使用连接参数[-gi]来保证该区域不被占用。

7. 地址 0xF7F0 到 0xF7FF

当使用伪实时 RAM 监视功能时，地址 0xF7F0 到 0xF7FF 用于设置软件断点。请注意该地址段不能被用户程序占用。

8. 堆栈指针设置值

用地址 0xFEDE 或者更低的地址作堆栈。

内部 RAM 区域根据 IMS 寄存器的设置有所不同，如果 RAM 从 0xFB00 开始，堆栈指针活动范围 0xFB07 到 0xFEDF(当使用软件断点时,堆栈指针范围为 0xFB09 ~0xFEDF；使用伪实时 RAM 监视功能，堆栈指针范围 0xFB10 ~0xFEDF)。堆栈大小可以根据功能不同而有所差别。地址 0xFEDF 不可以作为堆栈空间使用，但可以作为变量空间。MINICUBE 临时使用 0xFEDF 作为断点时程序处理的标志位，但是这个地址上原有的值会被保存。

9. OCD 过程中会导致 Flash 存储器内容改变的操作：

（1） 在下载对话框中操作
（2） 存储器填充对话框的操作
（3） 存储器拷贝对话框的操作
（4） 在存储器窗口修改
（5） 在汇编窗口在线汇编
（6） 程序执行时使用软件断点
（7） 在 Configuration 对话框中，“Target Power Off”参数选择“Permit”，程序执行中使用硬件断点或软件重启功能。
（8） 使用伪实时 RAM 监视功能

10. 自编程时的引导交换

自编程时使用的引导交换功能，是在簇单元中实现的，设备不同，簇地址不同。对于μPD78F0547D 而言，分为簇 0（0000H~0FFFH）和簇 1（1000H~1FFFH）。簇 0 包括了第 1 项为 MINICUBE 预留的 Flash 区域，这些区域内的数据都要拷贝到簇 1 中去，于是保留区域的地址变成（原地址+1000H）。

11. 自编程时的软件断点

当进行软件断点的设置时，调试器需要把原指令保存起来，在断点处将原指令替换成调试指令。因此，避免在将要被自编程覆盖的程序区域中设置软件断点；否则自编程过程完成后，调试器会将原来保存的指令回写到原指令所在的地址上，自编程所写入的内容会被覆盖。

12. 自编程模式 A1 时的仿真

通过自编程使用固件的功能时，模式就从正常切换到自编程 A1

```
MOV PFCMD,#0A5H
MOV FLPMC,#1H      ← <1>
MOV FLPMC,#0FEH
MOV FLPMC,#1H
----------------------- From this position
CALL !08100H    ← <2> ↑
MOV PFCMD,#0A5H
MOV FLPMC,#0H              Mode A1
MOV FLPMC,#0FFH
MOV FLPMC,#0H      ← <3> ↓
----------------------- To this position
```

〈2〉〈3〉之间是自编程模式 A1（包括 A2）。步进运行、断点都不能在这一段程序中执行，程序也不能在此处停止。而且在〈1〉〈3〉之间不要设置软件断点，否则不但断点不会起作用，还将使仿真不正常。

13. 堆栈指针初始化失败的中断

堆栈指针没有初始化，有断点发生或者有压栈指令操作时，则发生该中断。

如果 RESET 后马上执行下面的动作，那就可以避免这个错误的发生

- ◆ 设置一处软件断点
- ◆ 从寄存器窗口写入堆栈指针
- ◆ 从存储器窗口写 Flash Memory

如果堆栈指针的初始化完成前发生软件断点，状态栏将提示“Uninitialized Stack Pointer”，因此，必须确保 SP 的正确设置。

14. 下载 HEX 文件的注意事项

下载 HEX 文件时，如果有 0xFF 以外的数值被指定为填充数据，此时又使用了目标转换参数[-u]，就会和第 2 项[0x02,0x03 和 0x8F 以及更高地址的值]发生冲突，此时为了保持一致性，使用如下代码来解决，而不要使用连接参数[-go]。

注：代码比较长，限于篇幅，不在此列出。敬请参阅《NCT-MINICUBE78K0 On-Chip Debug Emulator User’s Manual》（U17029E）。

15. 软件断点

如果调试器因为软件断点的问题异常退出，被调试指令替代的指令就无法复原。为了防止这个错误，请在下载对话框选中“Object”参数。

16. “监控时钟”的设定

“System”选择的是“Monitor Clock”，那么设备的时钟源在 Break 时就由 MINICUBE 提供。如果外围仿真功能没有停止（“Peripheral Break”选择的是“Non Break”），因为时钟的改变，仿真时间可能就和实际运行有所差别。为了调整时钟的计数，应该在

MINICUBE 内部装一个和实际频率相同的振荡器。

17. 单步运行时的注意事项

单步运行时某些外围寄存器不动作。这时候，用“中断—运行”的办法来解决。

18. RAM 区域的软件断点

软件断点设置在内部低速 RAM 或者外部 RAM 时，避免覆盖已经设置了软件断点的程序，否则断点工作不正常。如果断点成功，则被覆盖的程序会自动恢复。所以，请在用户程序写入完成后再进行软件断点的设置工作。

19. 使用伪实时 RAM 监控功能的详细执行时间以及配置时的注意事项

请参阅《NCT-MINICUBE78K0 On-Chip Debug Emulator User’s Manual》（U17029E）。

20. 用于调试的设备如何处理

请不要将调试过程中使用的设备应用到批量生产的产品中。（Flash 在调试过程中已经被反复擦写，将来的写入过程就不能保证可靠性。）

21. Option Byte 的注意事项

当连接 MINICUBE 时，对于地址 0x80 到 0x83（地址根据设备的手册指定）区域的 Option Byte 必须被正确设置。

例子：

```
SSS    CSEG   AT   080H;     “SSS”是名称（最长八位）
       DB     0H
```

22. 片上 Flash 存储器安全功能

Flash 被写入之后，在系统对话框输入正确 Security ID 码进行校验才可以使用 ID78K0-QB for MINICUBE 进行调试。当 Flash 存储器中的内容被擦除后，那么 ID 密码就变回默认值 0xFFFFFFFFFFFFFFFFFFFF。

修订历史

版本号	修订页	修订内容	范围	修订日期
E1.00	1~81	文档新建	全文	25thJan,2006
V1.00	1~78	修改版	全文	6thFeb,2006
V1.00	41-54	结构调整	第六章	10thFeb,2006
V1.00	1-78	参考文档用全称	全文	10thFeb,2006
V1.00	II	加入版权说明	前言说明	10thFeb,2006
V1.00	74-79	删除 KX1+相关内容	附录 A，B	10thFeb,2006
V1.00	1-78	某些“编译”改为“Build”	全文	10thFeb,2006
V1.00	74-79	修改使用限制的说明	附录 A，B	10thFeb,2006
V1.00	1-78	“PM plus”改为“PM+”	全文	10thFeb,2006
V1.00	1-78	插图标注的对话框名称使用英文	全文	10thFeb,2006
V1.10	1-78	“QB78K0MINI”改为“NCT-MiniCube78K0”	全文	15thFeb,2006
V1.10	31,45,50	表格边框及文字调整	全文	15thFeb,2006
V1.10	32	删除 4.4 小节	第四章	15thFeb,2006
V1.10	34	图 5-2 标题由“窗体”改为“界面”	第五章	15thFeb,2006
V1.10	43,44	图 6-2,3,4,5 重新截图	第六章	15thFeb,2006
V1.10	48	插入特殊符号①②③	第六章	15thFeb,2006
V1.10	51	删除“FPL，KX1+”相关内容	第六章	15thFeb,2006
V1.10	58,69,71	重新截图	第七章	15thFeb,2006
V1.10	77	“PD78F0148H”—〉“PD78F0547D”	附录 B	15thFeb,2006
V1.20	39,40	加入 option.asm 和 Security ID 的详细说明	第五章	24thFeb,2006
V1.20	1~80	文档格式和用语	全文	2ndMar,2006